Efficient Dual-Process Cognitive Recommender Balancing Accuracy and Diversity

Yixu Gao[1], Kun Shao[3(✉)], Zhijian Duan[4], Zhongyu Wei[1,2(✉)], Dong Li[3], Bin Wang[3], Mengchen Zhao[3], and Jianye Hao[3]

[1] Fudan University, Shanghai, China
[2] Research Institute of Intelligent and Complex Systems, Fudan University, Shanghai, China
{yxgao19,zywei}@fudan.edu.cn
[3] Huawei Noah's Ark Lab, Beijing, China
{shaokun2,lidong106,wangbin158,zhaomengchen,haojianye}@huawei.com
[4] Peking University, Beijing, China
zjduan@pku.edu.cn

Abstract. In this paper, we propose a dual-process cognitive recommendation system for sequential recommendations. The framework includes an intuitive representation module (System 1) and an inference module (System 2). System 1 is designed to understand the user's historical interaction sequences with external knowledge graph. System 2 is built to make recommendations by reinforcement learning to consider long-term returns and diversity. We demonstrate the performance of our method on a wide range of recommendation datasets. Experiments show significant improvement over the state-of-the-art models regarding both relevance and diversity.

Keywords: Recommendation system · Knowledge graph · Reinforcement learning · Diversity in recommendation · Dual-process theory

1 Introduction

Traditional recommendation systems mainly follow the fashion of supervised learning. Methods including convolutional neural network [29] and graph neural network [26] are adopted to represent the item sequences and generate a probability distribution to support the recommendation. Researchers further propose attention mechanism [24], memory network [5], and hybrid model [21] for better representation learning of item sequence and obtain positive results. Recently, SANS [30] adds a neural similarity module and a setwise attention module to address the few-shot recommendation problem.

In order to deal of the scenario of dynamic sequential recommendations, which aim to recommend next-item or next-session according to historical interaction records, researchers formulate the task as sequential decision problem

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13247, pp. 389–400, 2022.
https://doi.org/10.1007/978-3-031-00129-1_33

and reinforcement learning (RL) is used to characterize each user's personalized preferences considering the long-term returns [23]. DRN [32] presents a complete framework, and DEERS [31] balances both the positive and negative feedback of users.

Various model-free techniques are proposed to improve the recommendation performance [3]. Model-based RL is also widely used to learn the user model [15]. Despite the great improvements the above works achieved, the training efficiency of RL is still an unavoidable problem [28].

In this paper, we also focus on dynamic sequential recommendation task, and we explore to use supervised learning technique to accelerate the training efficiency of RL. Inspired by the dual-process theory [7], we propose a dual-process cognitive recommendation system, $\mathrm{CogRec}_{\mathrm{Div}}$, for sequential recommendations. We first design an intuitive representation module (System 1) to simulate the cognitive process. Moreover, for the reason that the users with similar interactive behaviors may click on completely different items in the next action according to the behavior polymorphism, our proposed model takes the relevance-to-diversity balance issue into account, that is the inference process (System 2). Specifically, System 2 balances the issue of diversity while considering long-term returns with the support of RL and determinant point process (DPP) methods. Among them, RL is adopted to balance the long-term and short-term relevance of interactive items, while the DPP serves as the description of diversity since the determinant can express the degree of similarity between items. Experiments are designed from three aspects: accuracy, diversity, and training efficiency. The results show that our proposed model achieves an improvement of up to 4.1% on four datasets compared with current state-of-the-art models.

2 The Proposed Method

2.1 Framework

In order to balance the trade-off between the accuracy and the diversity of recommendation systems, we propose $\mathrm{CogRec}_{\mathrm{Div}}$, whose framework is shown in Fig. 1. Given an interactive sequence and the knowledge graph, the representation of the sequence is extracted by System 1. An intuition model is used to make sure that the necessary information of the recommendations is gained and gives the intuitive recommendations quickly. Then, the system recommend the information extracted by System 1 through System 2. In the recommendation process, the recommendation accuracy is improved through RL, and the recommendation diversity is enriched through the DPP.

2.2 Preliminary

The item sequence of a user u is denoted as $i^u_{m:n} \doteq \left(i^u_m, i^u_{m+1}, \ldots, i^u_{n-1}, i^u_n\right)$, where i^u_t is the t^{th} item in item set $\mathcal{I}$. m and n $(1 \leq m \leq n)$ are the start and end item respectively that the user u interacted with. Given a user u's interactive sequence

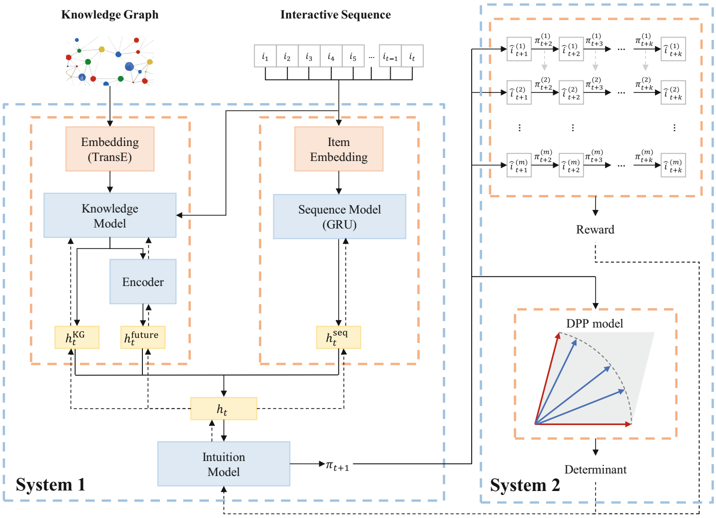


Fig. 1. The whole framework of our proposed method. The solid arrow represents the forward operation, and the dashed arrow represents the gradient back propaganda. System 1 efficiently extracts the intuitive representation from interactive sequence and the knowledge graph. Then, System 2 improves the accuracy by RL and enriches the diversity by DPP.

$i^u_{1:t}$, the goal is to predict the next k items $i^u_{t+1:t+k}$ that the user u would interact with. Each item i has its item embedding $e^{\text{item}}(i) \in \mathbb{R}^{d^{\text{item}}}$ and its knowledge graph embedding (KG embedding) $e^{\text{KG}}(i) \in \mathbb{R}^{d^{\text{KG}}}$, where $d^{\text{item}} \in \mathbb{Z}_+$ is the dimension of item embedding and $d^{\text{KG}} \in \mathbb{Z}_+$ is the dimension of KG embedding. For the sequence $i^u_{1:t}$, the item embedding sequence is $e^{u,\text{item}}(i_{1:t})$ and the KG embedding sequence is $e^{\text{KG}}(i^u_{1:t})$.

At each step t, $\mathbf{y}_t \doteq \left[y_t^{(1)}, y_t^{(2)}, \cdots, y_t^{(|\mathcal{I}|)}\right]$ is the user's ground-truth interactive vector, where $y_t^{(i)} (i = 1, 2, \cdots, |\mathcal{I}|)$ is the indicator function of the actions:

$$y_t^{(i)} = \begin{cases} 1, & \text{if the user interact with the item } i \text{ at time } t \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

2.3 System 1: Intuitive Representation Module

The extraction capacity of System 1 model is fundamental for recommendation, thus a powerful model is needed. Inspired by KERL [23], the representation h_t for an interactive sequence $i_{1:t}$ is consisted of three parts: sequence-level representation h_t^{seq}, knowledge-level representation h_t^{KG}, and future-level representation h_t^{future}. Specifically, h_t^{seq} is used to extract the sequence information of items; h_t^{KG} combines external knowledge; and h_t^{future} is for further inference.

The sequence-level representation h_t^{seq} is gained by a standard recurrent neural network:

$$h_t^{\text{seq}} = \text{GRU}\left(h_{t-1}^{\text{seq}}, e^{\text{item}}(i_t); \theta^{\text{GRU}}\right), \tag{2}$$

where GRU$(\cdot)$ is the Gated Recurrent Unit [6], and θ^{GRU} denotes all the related parameters of the GRU network. The average value of the KG embedding of the item sequence is used as the knowledge-level representation $h_t^{\text{KG}} = \frac{1}{t}\sum_{j=1}^{t} e^{\text{KG}}(i_j)$. And we use a neural network to get the future-level representation h_t^{future}

$$h_t^{\text{future}} = \text{MLP}\left(h_t^{\text{KG}}; \theta^{\text{MLP}_1}\right), \tag{3}$$

where MLP$(\cdot)$ is the Multi-Layer Perception, and $\theta^{\text{MLP}_i}(i = 1, 2, \cdots)$ denotes all the related parameters of the MLP network. The final representation h_t is the concatenation of three representation vectors:

$$h_t \doteq \left[h_t^{\text{seq}}, h_t^{\text{KG}}, h_t^{\text{future}}\right]. \tag{4}$$

In order to extract the information needed for recommendation, we first use supervised learning to learn sequence representation

$$\pi_{\theta^{\text{MLP}_2}}(\cdot|h_t) = \text{MLP}\left(h_t; \theta^{\text{MLP}_2}\right), \tag{5}$$

where $\pi_{\theta^{\text{MLP}_2}}(\cdot|h_t)$ is the probability distribution over the complete item set of the next item that the user may interact with at time t.

Cross entropy is adopted as the loss of the supervised learning:

$$\mathcal{L}^{\text{sys}_1} = \text{CE}\left(\pi_{\theta^{\text{MLP}_2}}(\cdot|h_t), \mathbf{y}_t\right), \tag{6}$$

where CE$(\cdot)$ denotes the cross entropy loss function.

2.4 System 2: Inference Module

System 2 is used to inference based on the information extracted by System 1 and make the recommendations. Two components are used in system 2. RL is used to improve the accuracy and consider the long-term benefits of recommendation and DPP is used to enrich the diversity of items.

A Markov Decision Process Formulation for Our Task. The recommender is an *agent* that interacts with the *environments*, which are the users. At each time step, the *state* is the hidden state h of the previous interacted items, and the *actions* are the recommended items. The *transition* function is the representation networks. The *reward* can be obtained from the user and the knowledge. The *policy* π is the probability distribution over all the possible actions.

Reinforcement Learning. Strategies learned by RL perform better in the long-term cumulative reward. The reward R is consist of two parts: the relative part R^{rel}, and the KG part R^{KG}.

$$R = R^{\mathrm{rel}} + R^{\mathrm{KG}}. \tag{7}$$

The relative reward is bulit by discounted cumulative gain (DCG), which is a metric widely used in recommendations $R^{\mathrm{rel}} = \mathrm{DCG}\left(\pi_t, \mathbf{y}_t\right)$, where the relative score is the probability of each item. To make the recommended items and the previous items be as close as possible on the knowledge graph, the reward from KG is the similarity between the embeddings of the recommended items and the previous items $R^{\mathrm{KG}} = \mathrm{Distance}\left(e^{\mathrm{KG}}(m_t), e^{\mathrm{KG}}(i_t)\right)$, where $m = \arg\max_{j\in\mathcal{I}} \pi_t^j$ is the item with the highest probability at step t, and here we use cosine similarity as the distance metric.

We use the truncated policy gradient as the RL algorithm, and the RL loss at time t is

$$\mathcal{L}^{\mathrm{RL}} = -\sum_{j=t}^{t+k} \gamma^{j-t} R_j \cdot \log p_i, \tag{8}$$

where γ is the discount factor, p_i is the probablity of the item that the user interacts with at the next step.

Determinantal Point Process. A DPP $\mathcal{P}$ on the whole item set $\mathcal{I}$ is a probabilistic model on $2^{\mathcal{I}}$, which is the set of all subsets of $\mathcal{I}$ [2]. When the empty set has nonzero probability, there exists a matrix $\mathbf{L} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ such that for every subset $C \subseteq \mathcal{I}$, the probability of C is $\mathcal{P}(C) \propto \det(\mathbf{L}_C)$, where $\mathbf{L}$ is a real, positive semidefinite (PSD) kernel matrix indexed by the elements of $\mathcal{I}$, and $\mathbf{L}_C$ is the sub-matrix of $\mathbf{L}$ indexed by C in both rows and columns. $\det(\mathbf{L})$ is the determinant of $\mathbf{L}$, and $\det(\mathbf{L}_\varnothing) = 1$ by convention.

Kernel Matrix Construction. The kernel matrix can be written as a Gram matrix, $\mathbf{L} = \mathbf{B}^\top \mathbf{B}$, where the columns of $\mathbf{B}$ are items representations. Each column vector $\mathbf{B}_i$ can be constructed by the product of the item score $r_i \geq 0$ and a normalized vector $\mathbf{f}_i \in \mathbb{R}^{d^{\mathrm{kernel}}}$ with $\|\mathbf{f}_i\|_2 = 1$, where $d^{\mathrm{kernel}} \in \mathbb{Z}_+$ is the dimension of item representation. Here, we use the normalized KG embedding as $\mathbf{f}_i = \mathrm{Norm}\left(e^{\mathrm{KG}}(i)\right)$, where $\mathrm{Norm}(\cdot)$ is the batch norm operation [12]. The item relevant score $r_{i,t}$ is constructed by the cosine similarity between the embeddings of the item i and the average of previous item embeddings before step t

$$r_{i,t} = \cos\left(e^{\mathrm{KG}}(i), h_t^{\mathrm{KG}}\right), \tag{9}$$

where $r_{i,t}$ is the relevant score of the item i at step t. The entries of kernel $\mathbf{L}_t$ is

$$\left(\mathbf{L}_t\right)_{i,j} = \langle \mathbf{B}_i, \mathbf{B}_j \rangle = \langle r_i \mathbf{f}_i, r_j \mathbf{f}_j \rangle = r_i r_j \langle \mathbf{f}_i, \mathbf{f}_j \rangle, \tag{10}$$

where $\langle \cdot, \cdot \rangle$ is the inner product of two vectors. The kernel matrix at step t is

$$\mathbf{L}_t = \mathrm{Diag}(\mathbf{r}_t) \cdot \mathbf{S} \cdot \mathrm{Diag}(\mathbf{r}_t), \tag{11}$$

where $\mathbf{S}$ is the similarity matrix of items that measures the distance between two items, and $\mathbf{S}_{ij} = \langle \mathbf{f}_i, \mathbf{f}_j \rangle$. Diag$(\cdot)$ is the diagonal matrix of the given vector.

Diversity with DPP. We hope that the final recommendation set can have as much diversity as possible. Therefore, we use the method of maximizing the determinant of the top k recommended items to achieve this goal

$$\mathcal{L}^{\text{DPP}} = -\det\left(\mathbf{L}_{t,D}\right), \tag{12}$$

where D is the item set with top k probability, which includes the top k items that the user will interact with next predicted by the recommendation system.

2.5 CogRec$_{\mathbf{Div}}$

We combine the two systems, System 1 and System 2 including RL and DPP parts together with different weights $\lambda_1, \lambda_2, \lambda_3$ to construct the final loss $\mathcal{L}$,

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}^{\text{sys}_1} + \lambda_2 \cdot \mathcal{L}^{\text{RL}} + \lambda_3 \cdot \mathcal{L}^{\text{DPP}}. \tag{13}$$

3 Experiments

We evaluate the accuracy, diversity, and efficiency of our proposed approach. We first describe the experimental settings.

3.1 Sequential Settings

Datasets. We conducted experiments on four datasets, including three e-commerce datasets, and a music recommendation dataset. We adopt three e-commerce categories from **Amazon** [17]: Books, Beauty, and CDs with different diverse sizes and sparsity levels. We take the subset of **Last.FM** [19] where the timestep is from Jan, 2015 to June, 2015. For all datasets, we remove users and items with less than three interaction records.

Evaluation Metrics. Following [23], we use Hit-Ratio@k and Normalized Discounted Cumulative Gain (NDCG@k) to evaluate the proposed method. Following [2], we use intra-list average distance (ILAD), and intra-list minimal distance (ILMD) to measure the diversity. Higher metrics are desirable. All the metrics were calculated based on the top-k recommendations to each user for each test case. k is set to 10. For each test case, we randomly sample 100 negative items and rank them with the ground-truth item.

Parameter Settings. We optimize all models with the Adam optimizer by setting the batch size as 2048, the coefficients used for computing running averages of gradient and its square as 0.001, the betas as (0.9, 0.999) without weight decay. The hidden layer sizes of the model used for Amazon dataset are set to 50, and 100 for Last.FM. All the hyper-parameters are obtained using grid search. The weights $\lambda_1, \lambda_2, \lambda_3$ are set to 1, 0.1 and 1 based on the analysis of experimental results. The path number and length are both set to 3 when sampling.

Table 1. Performance comparison between the baselines and our model. The best performance in each row is in bold font, and the starred numbers represent best baseline performance. The last column shows the absolute improvement of our results against the best baseline, which is significant at p-value ≤ 0.05.

Dataset	Metric ↑	KGAT	Ripple	GRU4Rec	KSR	KERL	$\text{CogRec}_{\text{Div}}$	Improvement
Beauty	Hit-Ratio@10	44.0	42.2	39.4	51.0	54.1*	**55.9**	3.3%
	NDCG@10	27.6	21.4	29.5	32.2	36.5*	**37.9**	3.8%
CDs	Hit-Ratio@10	63.4	58.4	50.5	68.3	73.7*	**75.1**	1.9%
	NDCG@10	41.7	37.6	32.9	45.0	50.8*	**52.6**	3.5%
Books	Hit-Ratio@10	70.2	63.8	56.2	75.1	80.0*	**80.7**	0.9%
	NDCG@10	45.8	42.8	38.5	52.4	57.1*	**58.7**	2.8%
Last.FM	Hit-Ratio@10	55.8	52.5	52.8	62.7	64.2*	**64.8**	0.9%
	NDCG@10	42.1	38.2	40.7	48.1	50.1*	**52.8**	5.4%

Compared Methods. Our model is compared with these competitive baselines:

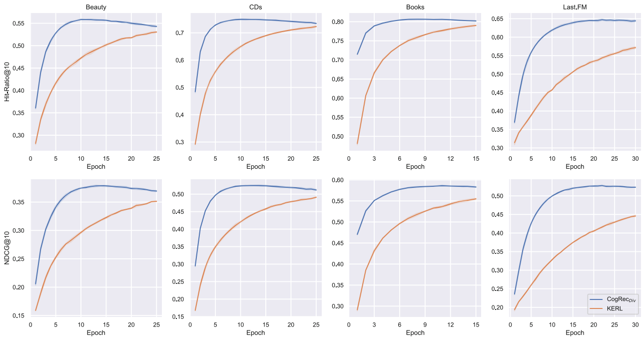
- **KGAT** [25] explores the high-order connectivity with semantic relations in collaborative knowledge graph for knowledge-aware recommendation.
- **Ripple** [22] models users' potential interests along links in knowledge graph for recommendation through embedding.
- **GRU4Rec** [10] utilizes GRU to capture users' long-term sequential behaviors in session-based recommendation.
- **KSR** [11] integrates a memory network to a GRU-based sequential recommender with Knowledge-Enhanced Memory Networks.
- **KERL** [23] explicitly discusses and utilizes knowledge graph information for the exploration process in sequential recommenders with RL.

For KERL, we use the released code. For other baselines, we followed the same settings and used the results in the KERL paper.

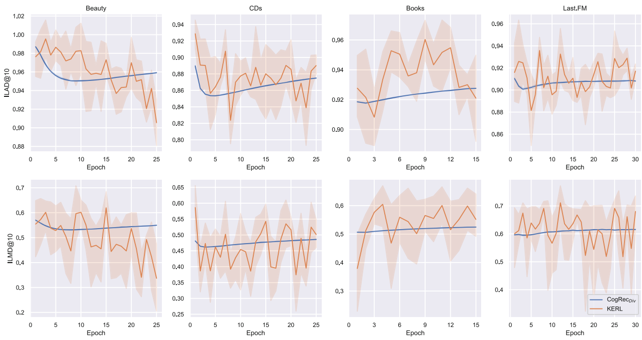
3.2 Main Results

Table 1 shows our evaluation results of recommendation models. On the four datasets, our model outperforms all baselines both with respect to the two metrics. On *Books* and *Last.FM*, although our model does not improve the Hit-Ratio@10 much, it improves NDCG@10 a lot. That is to say, although the top-10 items recommended by our model and the baselines all hit the next item in the user interaction, the target item of the user ranks higher in the list recommended by our model.

Figure 2(a) shows the training process of KERL and our model $\text{CogRec}_{\text{Div}}$. Our model is more efficient than KERL during training, and the training process is more stable. Figure 2(b) shows the diversity metrics through training process. The diversity of the items recommended by KERL without constrains of the determinant is very variate, while the diversity of the recommendations produced by $\text{CogRec}_{\text{Div}}$ is limited in a small range. Notes that the lowest points of the diversity curves appears earlier than the highest points of the accuracy curves, which means that the diversity is helpful to the improvement of accuracy.



(a) **Training metric**



(b) **Diversity metric**

Fig. 2. Accuracy and diversity metric curves in training process. An early stop signal has been triggered during the training process of the CogRec$_{\text{Div}}$ model shown by the blue line. (Color figure online)

3.3 Ablation Study

Recall that we have three losses in Eq. 13, namely $\mathcal{L}^{\text{sys}_1}$ (Eq. 6), $\mathcal{L}^{\text{RL}}$(Eq. 8), $\mathcal{L}^{\text{DPP}}$(Eq. 12). Therefore, we consider three variants for comparison by examining the effect of each part for sequential recommendation, including: (1) Rec using only the representation got from system 1; (2) CogRec using two systems, but only uses RL in system 2; (3) Rec$_{\text{Div}}$ using two systems, but only uses DPP in system 2.

Table 2 shows the results of CogRec$_{\text{Div}}$ model and its three variants on four datasets. In general, System 1 can extract the information needed for recommendation better than other baseline models. System 2 considering long-term returns and the diversity can effectively improve the accuracy of recommendations. And according to the substantial improvement of NDCG, it can be seen that apart from Hit-Ratio, our model can rank the items that users will click next in the recommendation list higher.

Table 2. Performance comparison of Rec, CogRec, $\mathrm{Rec}_{\mathrm{Div}}$ and $\mathrm{CogRec}_{\mathrm{Div}}$.

Dataset	Metric	Rec	CogRec	$\mathrm{Rec}_{\mathrm{Div}}$	$\mathrm{CogRec}_{\mathrm{Div}}$
Beauty	Hit-Ratio@10	55.7	55.9	55.8	**55.9**
	NDCG@10	37.6	37.7	37.9	**37.9**
CDs	Hit-Ratio@10	74.6	75.1	74.5	**75.1**
	NDCG@10	51.9	52.5	51.8	**52.6**
Books	Hit-Ratio@10	80.3	80.5	80.7	**80.7**
	NDCG@10	58.3	58.5	58.4	**58.7**
Last.FM	Hit-Ratio@10	61.6	64.5	61.4	**64.8**
	NDCG@10	50.9	52.5	50.9	**52.8**

For *CDs* and *Last.FM*, System 2 with RL (CogRec) performs better than System 2 with DPP ($\mathrm{Rec}_{\mathrm{Div}}$) in both Hit-Ratio@10 and NDCG@10. However, for *Books*, the diversity part contributes a lot in Hit-Ratio@10, which leads to the conclusion that the both parts in System 2 are effective in improving recommendation accuracy.

3.4 Case Study

Figure 3 presents an example of the items recommended by the models based on a user's interactive sequence consisting of 11 items. In the interactive sequence, nail polish accounted for nearly half of the user's clicks. The target item for user's next interaction is a dark purple nail polish, which is expected to be included in the recommendation lists and ranked as high as possible. Among the recommendation lists of the four models, only the Rec model fails to hit the target, which means that it is infeasible to predict this user's interactive preference based on simple intuition system. Fortunately, the recommended lists of the remaining three models all include the target item. Among them, $\mathrm{CogRec}_{\mathrm{Div}}$, which includes two complete systems, not only hits the target, but also ranks it first. This shows that considering the diversity of long-term returns, in this example, the recommendation accuracy is significantly improved.

Although Rec's recommendation failed to hit the target, the variety of recommendation was not low, which was consistent with the trade-off relationship of accuracy and diversity. In the recommendation list of CogRec with the long-term benefit (System 2), there are more nail polishes recommended than the other three ones. Correspondingly, the variety of items is relatively lower than others, which concludes that the model with RL is more inclined to recommend items that is similar to the previous ones. The outcome of $\mathrm{Rec}_{\mathrm{Div}}$ with System 2 has the highest recommendation diversity. It not only recommends items in previous categories, but also adds a lot of items in categories that have not appeared before, such as hair care and styling tools. $\mathrm{CogRec}_{\mathrm{Div}}$'s recommendations balance the effects of accuracy and diversity at the same time.

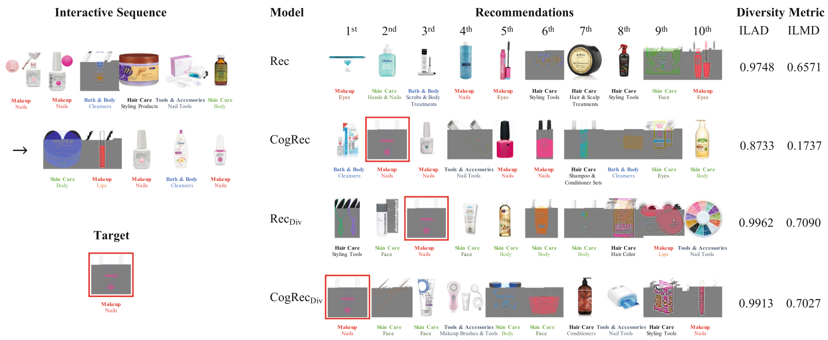


Fig. 3. A case where different models recommend items for the same sequence. The left side is the item sequence of the user's historical interaction and the target the user will click next. The middle one is the top 10 recommended items given by the four models. The category and subcategory of each item are marked below with bold and regular fonts. The diversity scores of each model's recommendation list are given on the right.

4 Related Works

The question of how the recommended satisfaction is affected by diversity and relevance has been deeply explored, and experiments show that diversity may have a negative net effect on user's satisfaction [13]. In terms of algorithms, the frequency of occurrence is considered, and at the same time, all projects are given fair exposure opportunities to strengthen the robustness of the system [18].

As for sequential recommendations, HISR [9] brings sequential dynamics into heterogeneous recommendations with textual and social information. SRecGAN [16] combines the integration of adversarial training with BPR. Some scholars have recently used the DPP [1] to improve the diversity of the recommendation system. The adoption of DPP overcomes the NP-hard calculation problem and greatly accelerates the greedy MAP inference for DPP, which has a better performance in the relevance-to-diversity trade-off [2,8].

Currently, there are some approaches that combines knowledge graph and RL together. PGPR [27] introduces a sampling multi-path policy gradient method, and Ekar [20] expresses the problem as a sequential decision-making process. KGTN [4] designs the original path to join methods, and CPR [14] adds an attention mechanism to the actor network and proposes an interactive path reasoning algorithm on a heterogeneous graph.

The closest work to ours is KERL [23], which is a sequential recommendation system that combines RL and knowledge graphs. Unlike KERL, our model considers the **diversity** of recommendations and avoids the lack of user interest capture caused by frequent recommendations of similar items. On the other hand, we adopt the method of RL to assist **supervised** learning label prediction. And through multi-task learning, we can achieve better results more efficiently.

5 Conclusions

In this paper, we construct a recommendation system $\mathrm{CogRec}_{\mathrm{Div}}$ based on the dual-process theory in cognitive science. The dual-process cognitive recommender divides the recommendation into two subsystems. System 1 is used to quickly give intuitive representation based on the user's historical interactive items and the KG. System 2 analyzes the information and make further inferences, considering the long-term cumulative returns with RL and enriching the diversity of recommendation with DPP. Experiments show that our model has achieved state-of-the-art performance on all four data sets, while ensuring the diversity, the long-term returns, and the learning efficiency.

References

1. Calandriello, D., Derezinski, M., Valko, M.: Sampling from a k-DPP without looking at all items. Adv. Neural. Inf. Process. Syst. **33**, 6889–6899 (2020)
2. Chen, L., Zhang, G., Zhou, E.: Fast greedy MAP inference for determinantal point process to improve recommendation diversity. In: Advances in Neural Information Processing Systems, vol. 31, pp. 5622–5633. Curran Associates, Inc. (2018)
3. Chen, S.Y., Yu, Y., Da, Q., Tan, J., Huang, H.K., Tang, H.H.: Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In: SIGKDD, pp. 1187–1196 (2018)
4. Chen, X., Huang, C., Yao, L., Wang, X., Liu, W., Zhang, W.: Knowledge-guided deep reinforcement learning for interactive recommendation. arXiv preprint arXiv:2004.08068 (2020)
5. Chen, X., et al.: Sequential recommendation with user memory networks. In: WSDM, pp. 108–116 (2018)
6. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP, pp. 1724–1734 (2014)
7. Evans, J.S.B.: Dual-processing accounts of reasoning, judgment, and social cognition. Annu. Rev. Psychol. **59**, 255–278 (2008)
8. Gan, L., Nurbakova, D., Laporte, L., Calabretto, S.: Enhancing recommendation diversity using determinantal point processes on knowledge graphs. In: SIGIR, pp. 2001–2004 (2020)
9. Geng, H., Yu, S., Gao, X.: Gated sequential recommendation system with social and textual information under dynamic contexts. In: Jensen, C.S., et al. (eds.) DASFAA 2021. LNCS, vol. 12683, pp. 3–19. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73200-4_1
10. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. CoRR abs/1511.06939 (2016)
11. Huang, J., Zhao, W.X., Dou, H., Wen, J.R., Chang, E.Y.: Improving sequential recommendation with knowledge-enhanced memory networks. In: SIGIR, pp. 505–514 (2018)
12. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML, pp. 448–456 (2015)
13. Kunaver, M., Požrl, T.: Diversity in recommender systems-a survey. Knowl.-Based Syst. **123**, 154–162 (2017)
14. Lei, W., et al.: Interactive path reasoning on graph for conversational recommendation. In: SIGKDD, pp. 2073–2083 (2020)

15. Liu, Q., et al.: Building personalized simulator for interactive search. In: IJCAI, pp. 5109–5115 (2019)
16. Lu, G., Zhao, Z., Gao, X., Chen, G.: SRecGAN: pairwise adversarial training for sequential recommendation. In: Jensen, C.S., et al. (eds.) DASFAA 2021. LNCS, vol. 12683, pp. 20–35. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73200-4_2
17. McAuley, J., Targett, C., Shi, Q., Van Den Hengel, A.: Image-based recommendations on styles and substitutes. In: SIGIR, pp. 43–52 (2015)
18. Miyamoto, S., Zamami, T., Yamana, H.: Improving recommendation diversity across users by reducing frequently recommended items. In: Big Data, pp. 5392–5394. IEEE (2018)
19. Schedl, M.: The LFM-1b dataset for music retrieval and recommendation. In: ICMR, pp. 103–110 (2016)
20. Song, W., Duan, Z., Yang, Z., Zhu, H., Zhang, M., Tang, J.: Ekar: an explainable method for knowledge aware recommendation. arXiv preprint arXiv:1906.09506 (2019)
21. Tang, J., et al.: Towards neural mixture recommender for long range dependent user sequences. In: The World Wide Web Conference, pp. 1782–1793 (2019)
22. Wang, H., et al.: RippleNet: propagating user preferences on the knowledge graph for recommender systems. In: CIKM, pp. 417–426 (2018)
23. Wang, P., Fan, Y., Xia, L., Zhao, W.X., Niu, S., Huang, J.: KERL: a knowledge-guided reinforcement learning model for sequential recommendation. In: SIGIR, pp. 209–218 (2020)
24. Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., Liu, W.: Attention-based transactional context embedding for next-item recommendation. In: AAAI, pp. 2532–2539 (2018)
25. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S.: KGAT: knowledge graph attention network for recommendation. In: SIGKDD, pp. 950–958 (2019)
26. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: AAAI, vol. 33, pp. 346–353 (2019)
27. Xian, Y., Fu, Z., Muthukrishnan, S., De Melo, G., Zhang, Y.: Reinforcement knowledge graph reasoning for explainable recommendation. In: SIGIR, pp. 285–294 (2019)
28. Xin, X., Karatzoglou, A., Arapakis, I., Jose, J.M.: Self-supervised reinforcement learning for recommender systems. In: SIGIR, pp. 931–940 (2020)
29. Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J.M., He, X.: A simple convolutional generative network for next item recommendation. In: WSDM, pp. 582–590 (2019)
30. Zhang, Z., Lu, T., Li, D., Zhang, P., Gu, H., Gu, N.: SANS: setwise attentional neural similarity method for few-shot recommendation. In: Jensen, C.S., et al. (eds.) DASFAA 2021. LNCS, vol. 12683, pp. 69–84. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73200-4_5
31. Zhao, X., Zhang, L., Ding, Z., Xia, L., Tang, J., Yin, D.: Recommendations with negative feedback via pairwise deep reinforcement learning. In: SIGKDD, pp. 1040–1048 (2018)
32. Zheng, G., et al.: DRN: a deep reinforcement learning framework for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp. 167–176 (2018)