World Wide Web

Reinforced KGs Reasoning for Explainable Sequential Recommendation

--Manuscript Draft--

Manuscript Number:	WWWJ-D-20-00446R1
Full Title:	Reinforced KGs Reasoning for Explainable Sequential Recommendation
Article Type:	Large-Scale Graph Data Analytics
Keywords:	Reinforcement learning; sequential recommendation; Path Reasoning; Knowledge graphs
Corresponding Author:	Shijun Liu Shandong University CHINA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Shandong University
Corresponding Author's Secondary Institution:	
First Author:	Zhihong Cui
First Author Secondary Information:	
Order of Authors:	Zhihong Cui
	Hongxu Chen
	Lizhen Cui
	Shijun Liu
	Xueyan Liu
	Guandong Xu
	Hongzhi Yin
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	We explore the semantic-rich structured information derived from the knowledge graphs (KGs) associated with the user-item interactions and aim to reason out the motivations behind each successful purchase behavior. Existing works on KGs-based explainable recommendations focus purely on path reasoning based on current user-item interactions, which generally result in the incapability of conjecturing users' subsequence preferences. Considering this, we attempt to model the KGs-based explainable recommendation in sequential settings. Specifically, we propose a novel architecture called Reinforced Sequential Learning with Gated Recurrent Unit (RSL-GRU), which is composed of a Reinforced Path Reasoning Network (RPRN) component and a GRU component. RSL-GRU takes users' sequential behaviors and their associated KGs in chronological order as input and outputs potential top-N items for each user with appropriate reasoning paths from a global perspective. Our RPRN features a remarkable path reasoning capacity, which is regulated by a user-conditioned derivatively action pruning strategy, a soft reward strategy based on an improved multi-hop scoring function, and a policy-guided sequential path reasoning algorithm. Experimental results on four of Amazon's large-scale datasets show that our method achieves excellent results compared with several state-of-the-art alternatives.

Revision Statement

A Letter to the EIC and AE

Dear Editor in Chief, World Wide Web journal

On behalf of our co-authors, I would like to appreciate you and the AE' valuable time and arrangement of reviewing this paper. Thank you for considering our paper could be accepted for publication in World Wide Web journal after minor modification. You and the other two reviewers' generous comments and suggestions make our paper more presentation-wise.

In this revised version, we have written a point-by-point letter to address the review comments carefully, as reflected in the following Section B.

Please let me know if there are any further comments.

Zhihong Cui

B Answer to comments from Reviewers

B.1 For Reviewer #1's comments

Point 1.
Figure 2 should be close to section 3.1, and it will make the reading more comfortable. As an alternative, notations in Table 1 can be shown earlier.

Thank you for your suggestion. We have made the following changes to the layout of the paper.
(1) We put Figure 2 immediately after section 3.1.
(2) We put Table 1 in section 2, which is corresponding to the concept definition and problem formulation and makes it easier to be understood.

We further reorganize the entire paper to ensure that the text description and its corresponding Figure or Table are on the same or adjacent pages.

Point 2.
The author needs to give a more detailed introduction in section 3.6 about their proposed algorithm. That would make the readers understand their algorithm more easily.

We appreciate your careful comments. We add a detailed introduction in section 3.6 about the proposed algorithm as follows.

In this section, we will explain our policy-guided sequential path reasoning algorithm, which can output the potential top-N items for each user with their reasoning paths from a global perspective. Its details are shown in Algorithm 1. It takes the user u, the policy network $\pi(\cdot\,|s,\tilde{A}_u)$, value network $\hat{v}(s)$, and the similarity threshold φ as input, and outputs a set of global reasoning paths P_K for each user with corresponding paths probabilities Q_K and paths rewards R_K^*. Each t-hop reasoning path $\hat{p} \in P_K$ ends with an item entity, which is regarded as one of the N final recommendation items.

The algorithm firstly calculates users' interests p(a) among all pruned actions $\tilde{A}_{k,t}$ in each sequential KG G_k^R, then it adds M actions $\{r_{k,t}, e_{k,t}\}$ with the highest probability interests in each step t to each reasoning path , thus we can obtain a temporary candidate reasoning path $P_{K,T}^{tmp}$ with corresponding paths generative probabilities $Q_{K,T}^{tmp}$ and path reward $R_{K,T}^{*tmp}$. However, all the candidate reasoning paths are optimal in each sub-graph G_k^R but not optimal in all of them. Thus, the algorithm recalculates the change of users' interests probabilities $\mathrm{p}(s_{k,T})$ in the terminal entity of each temporary path over time. A reward attention function is designed to calculate users' initial interests distribution $w_{k,T}$. Both users' interests probabilities $\mathrm{p}(s_{k-1,T})$ of the former period and interests distribution $w_{k,T}$ of the current period are put into GRU to output the interests probabilities $\mathrm{p}(s_{k,T})$ of the current period. To guarantee the diversity of reasoning paths, this algorithm sets a similarity threshold φ. The similarity of any two paths should be exceed φ. Otherwise, filter out one of the paths based on users' interests. Finally, all the reasoning paths $\hat{p}$ corresponding with their sequential paths probabilities $\mathrm{p}(s_{k,T})$ and paths rewards $w_{k,T}$ will be saved into the reasoning paths set P_K, paths probabilities set Q_K and paths rewards set R_K^*.

B.2 For Reviewer #2's comments

Point 1.

It would be better if the authors can discuss the time complexity of the main components.

We appreciate your careful comments and further discuss the time complexity of our RSL-GRU architecture in section 5.2.

In addition, the time complexity of our RSL-GRU architecture is superior or comparable to the baselines. As mentioned in section 3.2, we use a blocking strategy to build sequential KGs. There are K sub-graphs and each sub-graph contains b sub-block. Thus, the time complexity in building a sequential KG in each sub-block is much smaller than the whole KG building methods in the baselines, such as PGPR [33], KPRN [30], and KARN [47]. Denote the times in sub-block KG constructing as Ω, this option is conducted b times in all K sub-graphs. Since the concentration among sub-blocks, the time complexity of sequential KGs building is $\mathrm{T}(\Omega) = \mathrm{K} \times \mathrm{b} \times \Omega$. Then, our method uses a user-conditional derivatively action pruning strategy to find M actions in each step. Thus, the time complexity of this option grows exponentially along with the number of steps, which can be denoted as Ω^T and T is selected from {1, 2, and 3}. As described above, M is set at 250, which is way lower than the original action number. Compared with the baselines that save all actions [33], its calculation economizes a lot. The time complexity in the multi-hop path scoring function is $\mathrm{T}(\Omega) = \Omega^2$ according to formula (5) and the time complexity in the reward function is $\mathrm{T}(\Omega) = \Omega$ based on formula (7) and formula (8). We use the GRU in the final sequential modeling, its time complexity can be calculated by the product of input data and hidden layer and denoted as Ω^2. Above all, the worst and best time complexities of our RSL-GRU are Ω^3 and Ω respectively. In addition, its time complexity is much lower compared with FMG and CKE. Both PGPR and our model have a Ω^3 time complexity in the worst situation, but our model has a much lower calculation.

Although the time complexity of our model is a little higher in the worst situation than Ω^2 in DAN, KPRN, and KARN, its calculation is much smaller compared with them.

Point 2.

Datasets in section 5.1.1 need to be given the links or citations.

Thank you for your suggestion. We add a link (https://nijianmo.github.io/amazon/index.html) to the datasets in section 5.1.1 on page 13.

Noname manuscript No.
(will be inserted by the editor)

Reinforced KGs Reasoning for Explainable Sequential Recommendation

Zhihong Cui · Hongxu Chen · Lizhen Cui · Shijun Liu · Xueyan Liu · Guandong Xu · Hongzhi Yin

the date of receipt and acceptance should be inserted later

Abstract We explore the semantic-rich structured information derived from the knowledge graphs (KGs) associated with the user-item interactions and aim to reason out the motivations behind each successful purchase behavior. Existing works on KGs-based explainable recommendations focus purely on path reasoning based on current user-item interactions, which generally result in the incapability of conjecturing users' subsequence preferences. Considering this, we attempt to model the KGs-based explainable recommendation in sequential settings. Specifically, we propose a novel architecture called Reinforced Sequential Learning with Gated Recurrent Unit (RSL-GRU), which is composed of a Reinforced Path Reasoning Network (RPRN) component and a

Zhihong Cui
School of Software, Shandong University, Jinan,China
E-mail: czh@mail.sdu.edu.cn

Hongxu Chen
University of Technology Sydney, Australia
E-mail: hongxu.chen@uts.edu.au

Lizhen Cui
School of Software, Shandong University, Jinan,China
E-mail: clz@sdu.edu.cn

Shijun Liu
School of Software, Shandong University, Jinan,China
E-mail: lsj@sdu.edu.cn

Xueyan Liu
School of Computer Science and Technology, Jilin University, Changchun 130012, China
E-mail: xueyanl17@mails.jlu.edu.cn

Guandong Xu
University of Technology Sydney, Australia
E-mail: guandong.xu@uts.edu.au

Hongzhi Yin
The University of Queensland, Australia
E-mail: h.yin1@uq.edu.cn

GRU component. RSL-GRU takes users' sequential behaviors and their associated KGs in chronological order as input and outputs potential top-N items for each user with appropriate reasoning paths from a global perspective. Our RPRN features a remarkable path reasoning capacity, which is regulated by a user-conditioned derivatively action pruning strategy, a soft reward strategy based on an improved multi-hop scoring function, and a policy-guided sequential path reasoning algorithm. Experimental results on four of Amazon's large-scale datasets show that our method achieves excellent results compared with several state-of-the-art alternatives.

Keywords Reinforcement Learning · Sequential Recommendation · Path Reasoning · Knowledge Graphs

1 Introduction

As the semantic-rich information representation, KGs, which contains a large number of diverse entities and interactions in the real world, have achieved excellent capabilities in explainable recommendation [22,32]. On the one hand, the abundant entities in KGs are beneficial to excavate more abundant information for a superior recommendation. On the other hand, the various relations can be regarded as explicit interpretations among the entities, which endows the recommendation systems with potential explanation capabilities.

To date, much research on the KGs-based explainable recommendation are mainly divided into two streams. One is the KGs embedding based models [3], such as Trans Family methods [14,21], and the skip-gram based methods [16, 43]. These methods usually make a recommendation based on entities' similarity. Another stream is the path-based recommendation [18,19]. For example, a multi-constraint method [45] searches the "best fit" individualized learning path for learners. Both of these two modeling streams are valid and practical. However, we argue that the path-based approach features [11,12,28] are more potential for explicit reasoning and better explainability. Thus, in this work, we follow the path-based approach to expand the explainable recommendation with sequential modeling capacity.

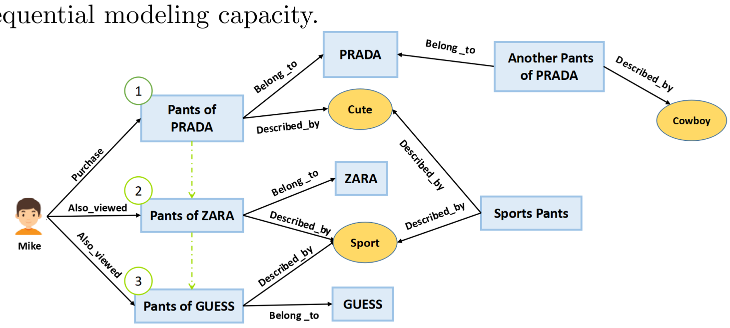


Fig. 1 An recommendation example based on a user's sequential historical behaviors with their associated KGs.

Although these two methods have achieved excellent explainable recommendation, they don't consider users' sequential historical behaviors. We argue that the sequence of users' historical behaviors can enhance the recommendation performance. Given an example in Fig. 1. Considering only Mike's first behavior, we can reasonably conjecture that Mike may like "Another Pants of PRADA" from the path "$\xrightarrow{Purchased}$ Pants of PRADA $\xrightarrow{Belong\_to}$ PRADA $\xrightarrow{-(Belong\_to)}$ Another Pants of PRADA". However, when considering Mike's historical behaviors' sequence: Pants of PRADA $\rightarrow$ Pants of ZARA $\rightarrow$ Pants of GUESS, we can rationally speculate what Mike really considered is something that has common features existed among all these three different brands rather than just another pair of pants from "PRADA". Thus, "Sport Pants" may be a more appropriate recommendation item than the "Another Pants of PRADA" since Mike's three historical behaviors all have a path to it. Several recommendation methods have been proposed from this perspective. For example, a knowledge-aware attentional reasoning network [47] predicts users' preferences by producing the representations of users' sequential historical interest and users' potential intent. An RNN-based network [30] leverages the sequence in one path. However, none of these methods has considered the KGs-based explainable recommendation as a sequential modeling issue.

However, there are several challenges to model KGs-based explainable recommendation as a sequential problem. Firstly, it is a formidable task. Current KGs-based recommendations aim to excavate KGs' abundant information in a spatial domain, while a sequential problem generally transforms features from a temporal perspective. Secondly, the measurement between the user and the terminal item in one path can not be easy since the relations between them are complicated. Thirdly, the size of the action space in KGs can run to millions. Hence, it is critical to design an efficient action-pruned method. Fourthly, a recommendation system must guarantee the diversity of reasoning paths since a model tends to trace actions and entities that have similar semantics with the previously positive samples.

To solve the first problem, we propose a Reinforced Sequential Learning with GRU architecture denoted as RSL-GRU in this paper. Specifically, it contains an RPRN and a GRU component to jointly search optimal items both in the spatial and temporal domain. To address the second problem, we propose an improved multi-hop scoring function. Although the multi-hop scoring function [33] can measure the relationship between users with terminal items, we argue that the user's preference for prior items can influence his subsequent choice. Considering this, we come up with an improved multi-hop scoring method. To deal with the third problem, we propose a user-conditional derivatively action pruning strategy to efficiently search promising actions in fixed action search space. To address the fourth problem, we come up with a policy-guided sequential path reasoning algorithm.

The major contributions of this paper are as follows:

– We propose a novel architecture called RSL-GRU to successfully model the KGs-based explainable recommendation as a sequential problem, which is driven by an RPRN and a GRU component.

- We design an RPRN to excavate information from KGs, which contains a soft reward function based on an improved multi-hop scoring strategy, a user-conditional derivatively action pruning strategy, and a policy-guided sequential path reasoning algorithm.
- We extensively evaluate the performance of our method on several Amazon e-commerce datasets in terms of accuracy recommendation and path reasoning. The results show the superiority of our method compared with state-of-the-art baselines.

2 Preliminaries

In this section, we introduce the concepts of the KGs and formulate the problem. Some important notations in this paper are summarized in Table 1.

2.1 Knowledge Graphs

Definition 3.1 (Knowledge Graphs) Formally, we establish the special KGs denoted as G^R, which consists of a series of segmented users' sequential items I with their associated KGs. It contains a subset of entities sets ε and a relation set R. The entities sets ε are composed of user entities U, a set of sequential items entities I, an associated entity set ε^*, where $U \cup I \cup \varepsilon^* \subseteq \varepsilon$ and $U \cap I = \phi$.

Definition 3.2 (t-hop path) a t-hop path is denoted as $p_t(e_0, e_t) = \{e_0 \overset{r_1}{\leftrightarrow} e_1 \overset{r_2}{\leftrightarrow} ... \overset{r_{t-1}}{\leftrightarrow} e_{t-1} \overset{r_t}{\leftrightarrow} e_t\}$, where $e_i \overset{r_{i+1}}{\leftrightarrow} e_{i+1}$ represents forward edge $e_i \xrightarrow{r_{i+1}} e_{i+1}$ or backward edge $e_i \xleftarrow{r_{i+1}} e_{i+1}$.

Definition 3.3 (t-hop pattern) a sequence of t relations for two entities is called a t-hop pattern (e_0, e_t) if there are a series of uniquely typed entities $e_1, \ldots e_{t+1}$. It can be formed by $\widetilde{r}_t = \{r_1, ..., r_t\}$.

Definition 3.4 (1-reverse t-hop pattern) a 1-reverse t-hop pattern is denoted by $\widetilde{r}_{t,j} = \{r_1, ..., r_j, r_{j+1}, ..., r_t\}$ $(j \in [0, t])$. Generally, $r_1, \ldots, r_j$ are forward, and $r_{j+1}, \ldots r_t$ are backward.

2.2 Problem Formulation

As users browse or purchase products every day, our KGs sequentially grow over time too. Then KGs from the first period to the last period k form a sequence $G^R_{1:K} = \{G^R_1, G^R_2, ..., G^R_k\}$, where G^R_k represents a series of users' sequential items I with its associated KGs in time k. So we can define our recommendation problem as follows.

Definition 3.5 (Reinforced Path-Reasoning Sequential Recommendation problem, RPRS-Rec) Given a series of sequential KGs $G^R_{1:K}$, the goal is to find a set of recommended items $\{i_n\}_{n \in [N]} \in I$, and give the reasoning path $p_t(u, i_n)$ between the user and the recommended items at the same time, where N is the number of final recommended items, T is the number of edges in each path, K is the number of segmented periods.

Table 1 Important notations

Notation	Description
U,u	user entities set U, $u \in U$
I,i	item entities set I, $i \in I$
ε, ε^*,e	entities set, $e \in \varepsilon$, $\varepsilon^* \in \varepsilon$
R, r	relations set R, $r \in R$
$G^R_{1:K}$	the dynamic KGs, consists of K KGs G^R
T,t	the number of steps or edges in a path
$\widetilde{r}_{t,j}$	t-hop pattern
$p_t\{e_0, e_t\}$	t-hop path
K	the number of segmented periods
N	the number of recommended items
h^*_t	the historical relations and entities prior to step t
S, s	state of entities, $s \in S$
$\widetilde{A}$	pruned action space
M	number of selected actions after pruned
R^*, r^*	reward set R, $r^* \in R^*$
P,p	path set, $p \in P$
Q,q	probability set, $q \in Q$
$\pi(\cdot\|s, \widetilde{A}_u)$	policy network
$\hat{v}(s)$	value network
O,o	Observation set, o is of each segmented period $o \in O$

3 RSL-GRU Architecture

In this section, we introduce the technical details of our RSL-GRU architecture.
3.1 Overall Structure

As a user's behaviors in e-commerce platforms are fast-changing, so do our KGs. Therefore, we build a sequential KGs-based model to globally generate top-N recommendations with their reasoning paths for each user. It mainly consists of three components: sequential KGs building, reinforced path reasoning using RPRN, and sequential modeling using GRU. Considering the complexity and long-tail distribution of KGs, we adopt a sequence of day-level KGs. For each period, we firstly to establish the KGs G^R_k based on the user's sequential behaviors and their associated KGs in chronological order. Here, to decrease the computation, we adopt a blocking strategy to divide each subgraph G^R_k into b sub-blocks G^R_{kb}. Then, we execute path reasoning on each block using a well-designed RPRN and integrate all excavated information of all blocks into a whole observation o^u_k of this period. Finally, we feed these sequential learned user's observations into a GRU network combined with an

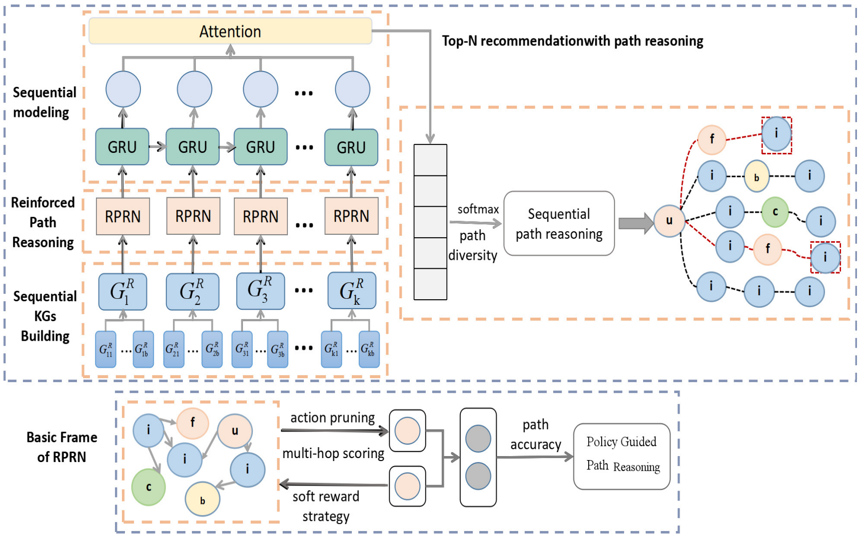


Fig. 2 The overall architecture of RSL-GRU for sequential KGs-based explainable recommendation.

attention mechanism to output the top-N items with appropriate reasoning paths. Fig. 2 shows the overall structure of our method.

3.2 Sequential KGs building using blocking strategy

Recall that the sequence of KGs plays a vital role in recommendation tasks. Considering this, we come up with a method to build a sequential KGs efficiently and effectively. Firstly, we sort all users' historical behaviors in chronological order. Then, we segment every three days of them into a period and there are K periods in all. Next, we build a subgraph for each period. In each period, according to the sequence of the user's historical behaviors, we extract corresponding entities and relations in KGs. Thus, we construct a new sequential KGs based both on users' historical behaviors and KGs' information for each period. However, this method will lead to a huge amount of calculation due to the following two reasons: (1) There are huge numbers of entities and relations in KGs, so the computation in each subgraph will be enormous; (2) We need to establish a subgraph for each period, which is repeated and abundant. Thus, we utilize the blocking strategy to establish each subgraph efficiently. For users' historical behaviors in each period, we rely on a divide-and-conquer strategy to partition the whole graph into non-overlapping sub-blocks $\{G_{k1}^R, ..., G_{kb}^R\}$ with the same size. Since each block is much smaller than the whole subgraph G_K^R, it would be faster if we excavate the KGs information in each block using the well-designed RPRN. It is noted that we ignore the relations between each block, which has been proved that it can effectively increase the computing speed with no loss to the final results [25]. As shown in Fig.2, we finally built a series of sequential subgraph G_k^R, each subgraph G_k^R is established by b sub-blocks G_{kb}^R.

3.3 Reinforced Path Reasoning using RPRN

The RPRN is a Reinforcement Learning (RL) model, which considers the information extraction problem as a Markov Decision Process(MDP)[26]. It firstly extracts node embeddings into a unified representation. Specifically, the agent in the RPRN starts from a user u and then obeys the guidance of the soft reward function to walk down along the pruned actions space $\widetilde{A}$ that pruned by the user-conditional derivatively action pruning strategy until it reaches the terminal entities e_t. In this process, the agent will record all possible paths P with their reward R^* driven by the policy-guided sequential path reasoning algorithm. After that, we can get the user' observation representation O_k^u of all periods KGs $G_{1:K}^R$. The details of our RPRN will be introduced in the next section.

3.4 Sequential Recommendation using GRU with attention mechanism

The user's observation o_k^u stands only for partial preferences, which couldn't sequentially speculate the user's preferences. Considering this, we model our RPRS-Rec problem as a sequential MDP.

As we all know, GRU always has an excellent performance in solving sequential problems due to its excellent ability to resolve the gradient vanishing problems. Thus, we adopt a GRU network here to recommend the final top-N items with reasoning paths. Specifically, it takes as input a sequence of embedding representations $O_k^u = \{o_1^u, o_2^u, \ldots, o_k^u\}$. Next, the hidden unit of GRU with an update gate z_k and a reset gate $\grave{r}_k$ controls the flow of information to select superior hidden states h_k from the candidate states $\widetilde{h}_k$. Afterward, the GRU network summarizes all observations O_k^u using a policy gradient conditioned on the user. It can be formalized as follows.

$$
\begin{aligned}
z_k^u &= \sigma(U_z o_k^u + W_z h_{k-1}^u + b_z) \\
\grave{r}_k^u &= \sigma(U_{\grave{r}} o_k^u + W_{\grave{r}} h_{k-1}^u + b_{\grave{r}}) \\
\widetilde{h}_k^u &= tanh(U_c o_k^u + W_c(r_k^u \odot h_{k-1}^u) + b_c) \\
h_k^u &= (1 - z_k^u) \odot h_{k-1}^u + z_k^u \odot \widetilde{h}_k^u
\end{aligned}
\tag{1}
$$

where $o_k^u \in R^d$is the input vector, $U \in R^{3\times d\times d}$ formed by U_z, $U_{\grave{r}}$ and U_c is the transition matrix for o_k^u, the logistic function $\sigma(x) = 1/(1 + e^{-x})$ is used to do non-linear projection, $\odot$ is the element-wise product between two vectors. $\widetilde{h}_k$ is the candidate state activated by element-wise $tanh(x)$. The output h_k is the current hidden state where k is the number of periods. To enhance the short-term interest in each hidden state, h_k^u contains not only information of the current period observation o_k but also critical information of the forgoing period h_{k-1}^{u}. In this way, the hidden units of GRU encapsulate the entire historical observations $o_{1:k}$ and output a sequence of hidden representation $\{h_1, h_2, ..., h_k\}$. Finally, we adopt softmax function to output the top-N items. To simplify, the RSL-GRU ignores the impossible newborn connection between two periods of observations.

Considering that different period's observation has different contributions to the final user's preferences recommendations, we adopt an attention mechanism to measure the importance. Specifically, we have the hidden representation of each period $\{h_1, h_2, ..., h_k\}$, the attention mechanism is shown as follows.

$$
\begin{aligned}
e_u^i &= q_u^T * h_i \\
a_u^i &= \frac{exp(e_u^i)}{\sum_{k=1}^{K} exp(e_u^k)} \\
o_u^{'} &= \sum_{k}^{K} a_u^k * h_i
\end{aligned}
\tag{2}
$$

where q^T is the attention vector, it's the sum of each user's reward in each period. $o_u^{'}$ is the final learned embedding of the user u. Thus, the rewards of each state in each period are also affected by the attention vector.

3.5 Optimization

As aforementioned, the sequential KGs-based explainable problem needs to be jointly optimized both in spatial and temporal domains.

The optimal goal of RPRN is to learn a policy to maximize the expected cumulative reward after multi-step for each user u. To solve this problem, we use a policy network $\pi(\cdot|s, \widetilde{A}_u)$ and a value network $\hat{v}(s)$ [33]. More specifically, the policy network $\pi(\cdot|s, \widetilde{A}_u)$ is designed to quantify the effect of each action on the current state s. It takes the current state s and pruned action space $\widetilde{A}(u)$ as input and emits the probability of each action, with zero for actions not in $\widetilde{A}(u)$. The value network $\hat{v}(s)$, which is the baseline in REINFORCE, is used to map the state s into real value. To minimize the error of the expected cumulative reward, we use Adam optimizer to train the RPRN. The optimal formula of the RPRN can be defined as follows.

$$
J(\theta) = E_{\pi}[sum_{t=0}^{t=T-1} \gamma^t R_{t+1}^{*} | s_0 = (u, u, \phi)] \tag{3}
$$

where γ^t is the discount factor at step t, R_{t+1}^{*} is the reward of step $t+1$, s_0 is the initial state. θ is the hyperparameters in those two networks.

From the optimal results of the RPRN, we can get the optimal value of the expected cumulative rewards between users with the terminal items after multi-step in each KGs period, which is defined as $g_k \in [0, 1]$. As mentioned above, these optimized values stand only for the optimal one in one segmented period. Thus, we here further adopt the GRU network to get globally optimal for each user. The optimal goal of the GRU network is to minimize the negative samples' effect. Specifically, we here employ the entities with non-zero rewards as positive samples and the remaining entities as negative samples. Thus, the loss function in the GRU network aims to maximize the following negative log-likelihood function.

$$L = -\{\sum_{y \in O^+} ylog(\widetilde{g}) + \sum_{y \in O^-} (1-y)log(1-\widetilde{g})\} \tag{4}$$

where O^+ are the positive samples, O^- are the negative samples.

3.6 Policy-guided Sequential Path Reasoning

In this section, we will explain our policy-guided sequential path reasoning algorithm, which can output the potential top-N items for each user with their reasoning paths from a global perspective. Its details are shown in Algorithm 1. It takes the user u, the policy network $\pi(.|s, \widetilde{A}_u)$, value network $\hat{v}(s)$, and the similarity threshold φ as input and outputs a set of global reasoning paths P_K for each user with corresponding paths probabilities Q_K and paths rewards R_K^*. Each t-hop reasoning path ends with an item entity, which is regarded as one of the N final recommended items.

The algorithm firstly calculates users' interests $p(a)$ among all pruned actions $\widetilde{A}_{k,t}$ in each sequential KG G_k^R, then it adds M actions with the highest probability interests in each step t to each reasoning path, thus we can obtain a temporary candidate reasoning paths $P_{k,T}^{tmp}$ with corresponding paths generative probabilities $Q_{k,T}^{tmp}$ and paths rewards $R_{k,T}^{*tmp}$. However, all the candidate reasoning paths are optimal in each sub-graph G_k^R but not optimal in all of them. Thus, it recalculates the change of users' interests probabilities $p(s_{k,T})$ for the terminal entity in each temporary path over time. A reward attention function is designed to calculate users' initial interests distribution $w_{k,t}$. Both users' interests probabilities $p(s_{k-1,T})$ of the former period and interests distribution $w_{k,t}$ of the current period are put into GRU to output the interests probabilities $p(s_{k,T})$ of the current period. To guarantee the diversity of reasoning paths, this algorithm sets a similarity threshold φ. The similarity of any two paths should exceed φ. Otherwise, filter out one of the paths based on users' interests. Finally, all the reasoning paths $\widehat{p}$ corresponding with their sequential paths probabilities $p(s_{k,T})$ and paths rewards $w_{k,t}$ will be saved into the reasoning paths set P_K, paths probabilities set Q_K and paths rewards set R_K^*.

Algorithm 1 Policy-Guided Sequential Path Reasoning

Require: u,$\pi(.|s, \widetilde{A}_u)$, $\hat{v}(s)$, similarity Threshold φ
Ensure: path set P_{T+1}, probability set Q_{T+1}, reward set R^*_{T+1}
Initialize: $P_0 \leftarrow \{\{u\}\}, Q_0 \leftarrow \{1\}, R^*_0 \leftarrow \{0\}$
for $k = 1$ to K **do**
  initialize $P_{k+1} \leftarrow \phi, Q_{k+1} \leftarrow \phi, R^*_{k+1} \leftarrow \phi$
  for $t = 1$ to T **do**
    initialize $P^{tmp}_{k,t} \leftarrow \phi, Q^{tmp}_{k,t} \leftarrow \phi, R^{*tmp}_{k,t} \leftarrow \phi$
    for all $a \in \widetilde{A}_{k,t}$ **do**
      Get path $\widehat{p}_{k,t-1}, s_{k,t-1} and \widetilde{A}_{k,t-1}(u)$ from environment
      $p(a) = \pi(a|s_{k,t-1}, \widetilde{A}_{u,k,t-1})$ and $a = (r_{k,t}, e_{k,t})$
      for $m = 1$ to M **do**
        $\widetilde{A}_{u,k} \leftarrow \{a|p(a) \in Top_M\}$
        Save the new path $\widehat{p} \cup \{r_{k,t}, e_{k,t}\}$ to $P^{tmp}_{k,t}$
        Save the new probability $p(a)\widehat{q}$to $Q^{tmp}_{k,t}$
        Save the new reward $R^*_{t-1} + r^*$ to $R^{*tmp}_{k,t}$
      end for
    end for
  end for
  Save $\widehat{p}$ if it ends with an item
  return $P^{tmp}_{k,T}$, $Q^{tmp}_{k,T}$,$R^{*tmp}_{k,T}$
  for all $\widehat{p}_{k,T} \in P^{tmp}_{k,T}$ **do**
    Get all $s_{k-1,T}, R^*_{k,T}$
    for all $s_{k-1,T}, R^*_{k,T} \in R^{*tmp}_{k,T}$ **do**
      $p(s_{k-1,T}) = v(s_{k-1,T})$
      $w_{k,T} = \frac{exp(\sum R^*_{k,T})}{\sum_1^k exp(\sum R^*_{k,T})}$
      $p(s_{k,T}) = GRU(w_{k,T}, p(s_{k-1,T}))$
    end for
    if $r^*_{diversity}(\widehat{p}, \widehat{p}^{'} \in P^{tmp}_{k,T}) > \phi$ **then**
      Save $\widehat{p}$ to P_K
      Save $p(s_{k,T})$ to Q_K
      Save $w_{k,T}$ to R^*_K
    end if
  end for
end for
return P_K, Q_K, R^*_K

4 Reinforced Path Reasoning Network

In this section, we introduce the detailed structure of RPRN. Its overall architecture is shown in Fig. 3. To better understand the model, we firstly introduce the improved multi-hop scoring function.

4.1 Improved multi-hop scoring function

The original multi-hop scoring function [33] only measures the relationship between the user and the terminal entity in a path. We argue that the user's preference for the terminal entity is affected by his former ones. Considering

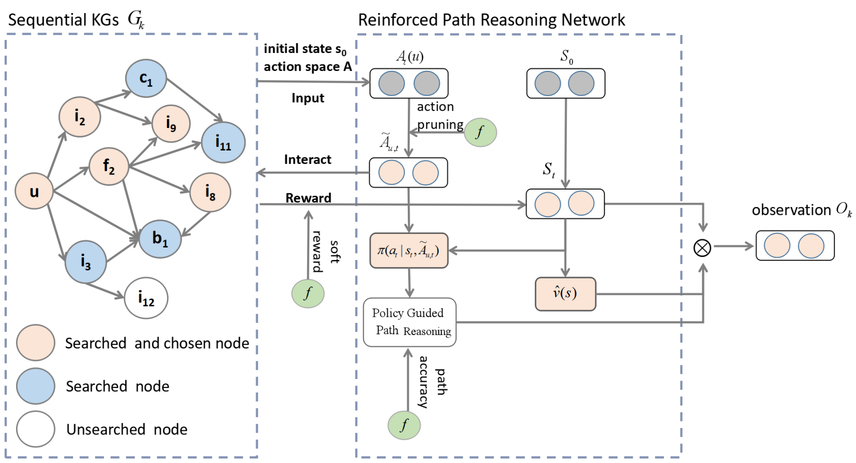


Fig. 3 The architecture of RPRN.

this, we propose an improved multi-hop score function.

$$f(e_0, e_t|\widetilde{r}_{t,j}) =< \frac{1}{j}(sum_{s=1}^{j}(e_0 + sum_{i=1}^{s} r_s)), \quad \frac{1}{t-s+1}(sum_{s=j+1}^{t}(e_t + sum_{i=j+1}^{s} r_s)) > +b_{et} \tag{5}$$

where <.,.> is dot operation, e, $r \in R^d, b_{et} \in R^d$ are d-dimensional vectors of the entities e and relations r and the bias of entity e. It calculates the relationship between the user and the terminal entity based on a cumulation of all preferences for the prior ones.

4.2 Components of RPRN

The RPRN contains a continuous state space S, an available action set $A = a_1, a_2, \ldots, a_n$, and a reward set R^*.

4.2.1 State

The state s_t is a tuple (u, e_t, h_t^*) at step t, where u is the starting user entity, e_t is the terminal entity the agent has reached after t steps, and $h_t^* = \{e_{t-k}, r_{t-k+1}, ..., e_t, r_t\}$ is the historical path prior to step t.

4.2.2 Action

The whole actions space contains all possible outgoing edges with their connected entities at state s_t except for the historical ones. Formally, the complete action space can be defined as $A_t = \{(r, e)|(e_t, r, e) \in G^R, e \notin \{e_0, \ldots ..e_{t-1}\}\}$. Since some entities' action space in the real-world can up to millions, it is inefficient and impractical to calculate all of them. Thus, we propose a user-conditioned action derivatively pruning strategy. Its principle will be introduced in the next section. The final pruned action space $\widetilde{A}$ is defined as follows.

$$\widetilde{A}_t(u) = \{(r,e)|len(rank(f((r,e)|u))) < M, (r,e) \in A_t\} \tag{6}$$

where M is the integer number of actions space after pruned, $f((r,e)|u)$ is the action scoring function, which is defined as a 1-reverse k-hop pattern with the smallest k using formula (5).

4.2.3 Reward

We propose the following scoring criteria to evaluate the paths.

Global accuracy: The global accuracy of a path dividing the user's selective probability on the terminal item e_t by the sum of the user's preferences for all items.

$$R^*_{GLOBAL_T} = \begin{cases} \frac{f(u,e_t)}{\sum f(u,i)} = \frac{f(u,e_t)}{\sum f(u,i|\widetilde{r}_{1,1})}, & i \in I and e_t \in I \\ 0, & otherwise \end{cases} \tag{7}$$

where $e_t = \sum_{t=1}^{t=T} r_t$ represents the path embedding for the relation chain $r_1 \rightarrow r_2 \rightarrow ... \rightarrow r_T$.

Path diversity: A recommendation system with excellent explainability should provide diverse reasoning paths. Hence, we define a diversity reward function as follows.

$$R^*_{DIVERSITY_T} = -\frac{1}{|F|}\sum_{i=1}^{|F|} cos(\widetilde{r}, \widetilde{r}_i) \tag{8}$$

where F is the number of existing paths, $\widetilde{r} = \{r_1, r_2, ..., r_t\}$ is the relation embedding for the path.

4.3 User-conditioned Action Derivatively Pruning Strategy

The basic principle is as follows. It firstly chooses a user as the initial state and then maps every connected action (r,e) to a real-valued score conditioned on the user. Next, it chooses M actions with the highest scoring as the start entities of the next step and repeats the above operations until the final step T.

Take the sequential KGs part of Fig.3 as an example. Supposing choosing two candidate actions in a two-step path, the agent starts from u and calculates the scoring of its all neighbor actions, such as i_2, f_2, b_1, i_3. Supposing that i_2, f_2 have the top two highest scorings, thus they are chosen as the start of next step and stored into the current state s_1. Repeat the above process until the terminal step. Through this strategy, the calculation complexity is fixed in a certain quantity as the step grows rather than exponentially growing in PGPR.

5 Experimental Evaluation

In this section, we extensively evaluate the performance of RSL-GRU architecture on real-world datasets.

5.1 Experiments Setup

In this section, we apply our RSL-GRU method on the following four Amazon datasets to evaluate its performance in different domains. We firstly introduce the datasets and baselines briefly. Then, we design several experiments aiming to address the following research questions:

- RQ1. How does RSL-GRU perform in top-K recommendation compared with the baselines?
- RQ2. What is the influence of improved scoring function?
- RQ3. What is the impact of user-conditioned derivatively action pruning strategy?
- RQ4. What is the influence of attention mechanism?
- RQ5. How does RSL-GRU perfom in terms of explainability?

5.1.1 Datasets

We apply our RSL-GRU method on the following four widely used Amazon e-commerce datasets[1] from different domains to evaluate its performance, such as *Beauty*, *Clothing*, *Books*, *Movies&TV*. Each dataset consists of both users' behaviors and meta information. Here, we firstly deleted the users whose clicked items are fewer than 3. Then, we sort the remaining users' behaviors by time-stamp. These datasets span from May 1996 to July 2014. We argue that behaviors from a long time ago make no sense for the users' recent preference recommendation. Thus, we only randomly sample users' latest three months behaviors in each dataset to predict the top-N recommendation items. In average, we selected 91,946, 85,130, 97,950 and 59,000 users' behaviors in *Beauty*, *Clothing*, *Books*, *Movies&TV*, respectively. Then, each dataset is segmented into 30 periods and each period contains users' three days level sequential behaviors. Considering the long-tail distribution in KGs, we then adopt Term Frequency-Inverse Document Frequency (TF-IDF) to prune the relations with less prominent features and keep the frequency of feature words less than 5,000 with TF-IDF score > 0.1. Finally, the users' behaviors are divided into training and testing sets of 30 % and 70 %, respectively.

5.1.2 Baselines

We compare our method with the following state-of-the-art baselines.

[1] https://nijianmo.github.io/amazon/index.html

- FMG (Factorization Machine Group with lasso) [40] is a meta-path based model that employs a factorization machine to assemble user or item vectors for rating recommendation.
- CKE (Collaborative Knowledge-based Embedding) [37] is a modern neural recommendation system to infer the top-N recommendations based on auxiliary information.
- DAN (Deep Attention-based Network) [46] uses an attention mechanism to extract users' features from their history clicked sequence for a recommendation.
- PGPR [33] utilizes an RL model for recommendation items and reasoning paths at the same time.
- KPRN (Knowledge aware Path Recurrent Network) [30] It's a KGs-based path recurrent network, which can well infer the rationale of user-item interaction based on the well-designed path representation and a weighted pooling operation.
- KARN (Knowledge-aware Attentional Reasoning Network) [47] incorporates the users' clicked history sequences and path connectivity between users and items for recommendation.

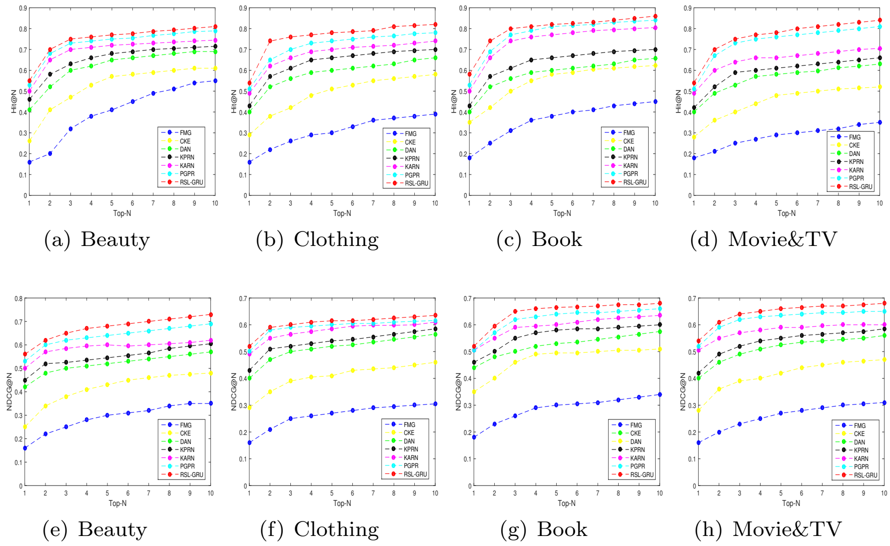
5.1.3 Parameter Setting

The default parameter settings in all experiments are as follows. The path length in our method ranges from 0 to 3. For sequential KGs building, all entities e_t and relations r are embedded into a 100-dimension vector, and the historical path h_t^* is a concatenation of entities and relations. The relations are embedded bidirectionally. Besides, we set $M = 250$ actions at each state. Furthermore, we divide the subgraph of each period into 20 blocks. In RPRN, we train the model 500 epochs using Adam optimization. Besides, we set a learning rate of η of 10^{-2} and a batch size of 64 for all datasets. The discount factor γ is 0.99. In the process of sequential modeling, we set a ratio between positive and negative interaction at 1:100, namely, 100 negative items are randomly sampled and pair with one positive item. In each GRU, we set the learning rate at 10^{-1} and the batch size at 64 for all datasets. We train our model 500 epochs using Adam optimization. The weight of the entropy loss is 0.001. To fairly compare, all the baselines are rerun based on a 1-hop scoring function shown in formula (5). All recommendation models are evaluated by the Normalized Discounted Cumulative Gain (NDCG) ($NDCG@N$) and the Hit Ratio (HR) ($Hit@N$) at Rank N.

5.2 RQ1. Performance Comparison

In this section, we evaluate the performance of our model on four datasets compared with several state-of-the-art baselines on the top-N recommendation. All the experiment results are shown in Fig. 4.

As shown in Fig. 4, our model RSL-GRU outperforms other baselines on four datasets with all metrics. More specifically, RSL-GRU increased by an



(a) Beauty (b) Clothing (c) Book (d) Movie&TV

(e) Beauty (f) Clothing (g) Book (h) Movie&TV

Fig. 4 Recommendation effectiveness of our model compared with baselines on Hit@N and ndcg@N

average 1.4 %, 2.3 %, and 2.8 % over PGPR, KARN, and KPRN, respectively, in terms of $(Hit@N)$. When it comes to $(NDCG@N)$, it achieves at least 2.5 %, 0.8 %, 1.2 % and 3.1 % higher performance than other models in $Beauty$, $Clothing$, $Books$, $Movies\&TV$, respectively. According to our research, there are three reasons that make its superiority of recommendation performance: (1) We conducted user-conditioned recommendations based on the users' historical behaviors associated with their KGs.(2) We well designed an RPRN to excavate the rich information from KGs, which can not only obtain the relation and items conditioned on the user but also can conduct diversified path reasoning. (3) We use a GRU network with an attention mechanism to further selectively learn users' preferences from a sequential perspective.

More details are as follows: (1) In Fig. 4, the meta-path-based method FMG has the worst performance among all the baselines. It just hit users' preferences at 3%. It is mainly because this method explores the entities and relations only based on the predefined meta-paths, which may lead to an information gap outside the set paths. (2) Both CKE and DAN perform much better than FMG. According to our research, both of them utilized the rich auxiliary information, which can indirectly prove the effectiveness of mining more information in the KGs. (3) DAN has a better performance than CKE in our experiments, and KARN is also better than KPRN. The main reason is that the attention mechanism can help DAN and KARN capture more reliable information, which gives a piece of explicit evidence that the attention mechanism in our model may also be capable of learning users' behaviors and associated KGs more effectively. (4) Except for our model, other sequential methods, e.g. KPRN and KARN, perform much better than the general methods (FMG, DAN, CKE). It indicates that the sequential features with KGs information can better explore the user-item interactions to infer users' preferences. (5) Among all the baselines, the RL-based method PGPR, which has an effective path reasoning process based on the well-designed KGs excavation policy, has the best performance. It indicates that a policy-guided path reasoning process can well explore the abundant information in KGs.

In addition, the time complexity of our RSL-GRU architecture is superior or comparable to the baselines. As mentioned in section 3.2, we use a blocking strategy to build sequential KGs. There are K sub-graphs and each sub-graph contains b sub-block. Thus, the time complexity in building a sequential KG in each sub-block is much smaller than the whole KG building methods in the baselines, such as PGPR [33], KPRN [30], and KARN [47]. Denote the times in sub-block KG constructing as Ω, this option is conducted b times in all K sub-graphs. Since the concentration among sub-blocks, the time complexity of sequential KGs building is $T(\Omega) = K \times b \times \Omega$. Then, our method uses a user-conditional derivatively action pruning strategy to find M actions in each step. Thus, the time complexity of this option grows exponentially along with the number of steps, which can be denoted as Ω^T and T is selected from {1, 2, and 3}. As described above, M is set at 250, which is way lower than the original action number. Compared with the baselines that save all actions[33], its calculation economizes a lot. The time complexity in the multi-hop path scoring function is $T(\Omega) = \Omega^2$ according to formula (5) and the time complexity in the reward function is $T(\Omega) = \Omega$ based on formula (7) and formula (8). We use the GRU in the final sequential modeling, its time complexity can be calculated by the product of input data and hidden layer and denoted as Ω^2. Above all, the worst and best time complexities of our RSL-GRU are Ω^3 and Ω respectively. In addition, its time complexity is much lower compared with FMG and CKE. Both PGPR and our model have a Ω^3 time complexity in the worst situation, but our model has a much lower calculation. Although the time complexity of our model is a little higher in the worst situation than Ω^2 in DAN, KPRN, and KARN, its calculation is much smaller compared with them.

5.3 RQ2. Impact of Improved Multi-Hop Scoring Function

We argue that a shorter reasoning path is more efficient on the reasoning, but a certain amount of steps may provide more reliable information. Thus, we evaluate the performance of our model under different hop with $hop = \{1, 2, 3\}$. To illustrate the effectiveness of our method, we use PGPR as our baseline because it uses an original multi-hop function. To fairly verify the impact of our improved multi-hop scoring function, we set our model the same as PGPR except for the different multi-hop scoring function. Besides, the experiments are measured by $Hit@10$ and $NDCG@10$ under the four datasets. The experiment results are shown in Table 2.

Table 2 Performance comparison under different hop size with $Hop = \{1, 2, 3\}$

Hit@10	Beauty			Clothing			Book			Movie&TV		
	1	2	3	1	2	3	1	2	3	1	2	3
PGPR	0.380	0.730	0.724	0.432	0.750	0.743	0.502	0.802	0.795	0.395	0.760	0.754
RSL-GRU	0.413	0.741	0.736	0.451	0.763	0.758	0.512	0.826	0.819	0.404	0.773	0.767
NDCG@10	1	2	3	1	2	3	1	2	3	1	2	3
PGPR	0.331	0.642	0.638	0.201	0.543	0.537	0.327	0.631	0.625	0.342	0.65	0.643
RSL-GRU	0.355	0.661	0.656	0.217	0.560	0.553	0.335	0.651	0.643	0.351	0.667	0.661

As shown in Table 2, our method outperforms PGPR on four datasets with all metrics. More specifically, our improved multi-hop scoring function can achieve at least 3% and 2% higher performance than PGPR on Hit@10 and $NDCG$@10, respectively. The following advantages of our improved multi-hop scoring function make its outstanding performance. Our multi-hop scoring function can measure the relevancy through the global paths between the initial user and the terminal item rather than just the beginning and final entities. It means that even if the initial user and terminal item of the two paths may be the same, their relevancy may be different. Thus, the average value of the different paths is more accurate than just direct relevancy. In summary, our improved multi-hop function can provide a recommendation with more outstanding performance than the original one.

Besides, here are other impressive experimental results: (1) Among all the datasets, both our model and PGPR with 2-hop and 3-hop perform superior to 1-hop under all metrics. It depends mainly on the multi-hop function, which can effectively capture the relevancy between entities with longer paths. (2) Both two models with 2-hop are further improved than that with 3-hop. In terms of Hit@10, the performance of our model and PGPR with 1-hop achieves at least 0.2% higher than these with 3-hop. The reason may be that longer paths may mislead the path reasoning process. (3) All models under 1-hop have a poor recommendation performance. It is because the entities with less information is not sufficient for an agent to search the related recommendation items.

5.4 RQ3. Impact of User-conditioned Derivatively Action Pruning Strategy

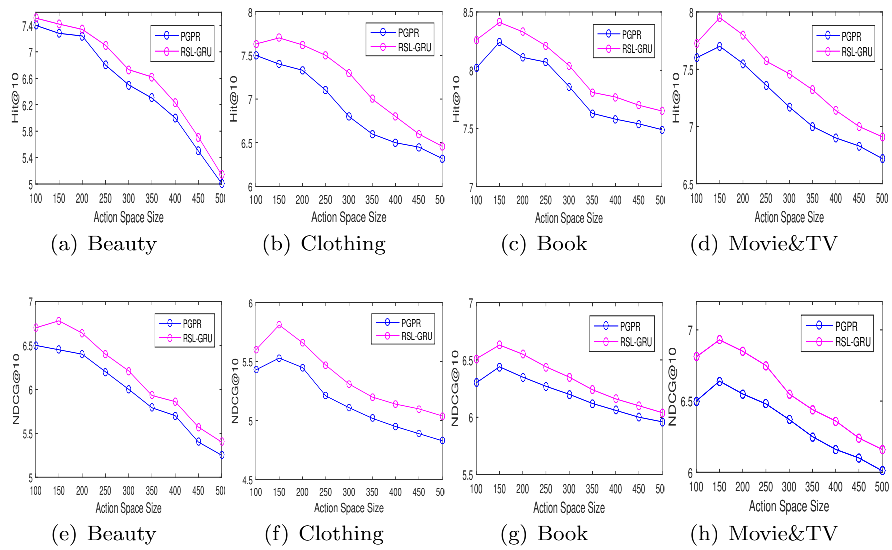
In this section, we evaluate the performance of our model on four datasets under different action space $\widetilde{A} = \{100, 150, 200, ..., 500\}$ to illustrate the impact of our user-conditioned derivatively action pruning strategy. Since PGPR is the only method with an original action pruning strategy among all these baselines, we compare our method with it. To fairly compare, we just set our model the same as it except for a user-conditioned derivatively action pruning strategy. Both of them are conducted in one-hop. Besides, we measured them under Hit@10 and $NDCG$@10. All experiment results are shown in Fig. 5.

As shown in Fig. 5, our user-conditioned derivatively action pruning strategy has better performance than PGPR on four datasets with all metrics. More specifically, our well-designed action pruning strategy can achieve at least 0.5

higher performance on Hit@10 and $NDCG$@10. According to our research, the main reasons are as follows: (1) Benefit from the improved multi-hop scoring function, our user-conditional derivatively action pruning strategy is capable of re-evaluating the current choice by comprehensively considering the entities in the whole path. Thus, it ensures a high correlation between the initial users and the terminal items. (2) We execute the user-conditioned action pruning strategy at each step, while PGPR only searches a certain number of actions initially. (3) Different from randomly sampling fixed quantity actions in PGPR, our model maintains a moderate number of actions with the highest scoring at each step.

Generally speaking, the model under both action pruning strategies shows a downward trend as we gradually increase the action space size. The reason is as follows. Although bigger action space means more available information, it also means there may be more redundancy and useless interference information. For instance, there are a large number of redundant relationships in $Beauty$, such as $Described\_by$ and $Mention$, which may cause information disorder. Thus, these two lines are both decreasing rapidly due to the increase of action space size on $Beauty$.

In conclusion, the recommendation system with our user-conditioned derivatively action pruning strategy can achieve outstanding performance under most action space size. Besides, we also find that a small action space is helpful for better performance.



(a) Beauty (b) Clothing (c) Book (d) Movie&TV

(e) Beauty (f) Clothing (g) Book (h) Movie&TV

Fig. 5 The recommendation performance under different sizes of action space compared with PGPR on Hit@N and NDCG@N

5.5 RQ4. Impact of Attention Mechanism

In this section, we evaluate the impact of attention mechanism on four datasets under $Hit@10$ and $NDCG@10$. In particular, we disable the attention mechanism as shown in (2), and rename it as RSL-GRU-0. For a fair comparison, we set all the rest of the parameters the same. Finally, we summarize the experimental results in Table 3 and have the following conclusions:

- The attention mechanism does have a positive effect on our model, which at least achieves 0.3 and 0.2 higher performance in terms of $Hit@10$ and $NDCG@10$, respectively. One main reason is that the items that users may choose in each period time might have different influence factors on the final recommendation. If we treat all period observations equally, it might mislead the sequential recommendation process.
- The attention mechanism has a different improvement on each dataset. In particular, it achieves the best improvement on the $Movie\&TV$ dataset in both metrics. After searching, we find that the users' historical behaviors vary greatly in the $Movie\&TV$ dataset. Thus, if we give them different weights, the model learned by our method is more in line with the real situation.

Table 3 The effect of attention mechanism on our model in four datssets

Metrics	Methods	Beauty	Clothing	Book	Movie&TV
Hit@10	RSL-GRU-0	0.787	0.797	0.839	0.805
	RSL-GRU	0.817	0.825	0.863	0.841
NDCG@10	RSL-GRU-0	0.705	0.612	0.651	0.648
	RSL-GRU	0.734	0.635	0.679	0.683

Table 4 Performance comparison in finding valid paths per user, unique items per user and paths per item compared with baselines

Valid Paths/User	Beauty	Clothing	Book	Movie&TV
KPRN	52.78 ± 5.96	53.35 ± 6.88	127.19 ± 13.95	102.84 ± 12.76
PGPR	59.95 ± 6.28	60.78 ± 7.00	153.25 ± 21.78	126.71 ± 13.19
RSL-GRU	67.49 ± 6.21	67.93 ± 6.84	177.28 ± 22.35	155.51 ± 17.92
Items/User	**Beauty**	**Clothing**	**Book**	**Movie&TV**
KPRN	34.15 ± 6.93	33.79 ± 7.04	103.17 ± 10.74	57.79 ± 8.39
PGPR	36.91 ± 7.24	37.21 ± 7.23	115.75 ± 12.63	68.26 ± 12.94
RSL-GRU	40.72 ± 7.03	40.76 ± 7.12	123.35 ± 27.19	80.35 ± 13.21
Paths/Item	**Beauty**	**Clothing**	**Book**	**Movie&TV**
KPRN	1.54 ± 1.03	1.58 ± 1.07	1.23 ± 1.13	1.78 ± 1.27
PGPR	1.62 ± 1.25	1.63 ± 1.25	1.32 ± 1.25	1.85 ± 1.31
RSL-GRU	1.66 ± 1.17	1.68 ± 1.20	1.44 ± 1.21	1.93 ± 1.52

5.6 RQ5. Explainability Comparison

All the experiments above show that our RSL-GRU model has an excellent recommendation performance. Still, beyond that, another desirable property of our RSL-GRU model is to reason on paths.

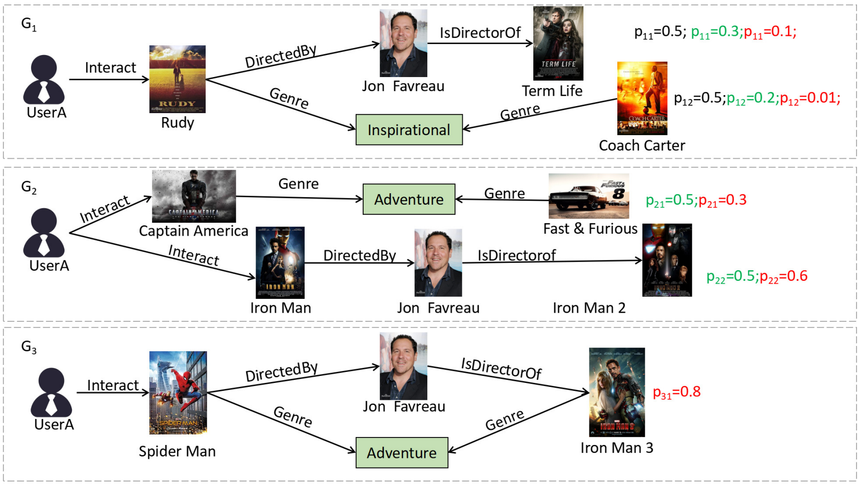


Fig. 6 Case Study for Path Reasoning

To evaluate the explainability of our method, we first measure its ability to find valid reasoning paths. We argue that a recommendation with excellent explainability should provide more valid reasoning paths. We randomly sample 125 valid paths for *Beauty* and *Clothing* datasets, and 200 for the other two datasets. To fairly compare, we use PGPR and KPRN as our baselines since both of them can reason on paths to generate reasonable explanations. All experiment results are shown in Table 4. Generally speaking, our method can find approximately 0.69 of the valid paths for each user, which is increased by 0.11 and 0.19 compared with the PGPR and KPRN, respectively. Besides, each item is endowed with 1.7 paths on average. It means that our method can provide multiple reasoning paths as interpretations. The two advantages of our RPRN make its outstanding recommendation performance: (1) We take into account users' historical behaviors and their associated KGs information to speculate on users' preferences, which implies that our method can sequentially excavate users the optimal recommendation items in richer and diverse choices. (2) The RPRN architecture is equipped with a superior path reasoning capacity due to its well-designed path reasoning policy.

Secondly, we show several cases generated by our model on the sequential explainable task in the *Movie&TV* dataset. Besides, we also use different colors to indicate recommended products at each period times: black for first, green for a second, red for third. In this experiment, we set path steps at 3. As shown in the first period time G_1 in Fig. 6, the user interacts with a movie called "Rudy". Next, our method finds two paths in KGs: it is an inspirational movie and directed by "Jon Favreau". From these two perspectives, our model recommends "Coach Carter" and "Term Life" both in 0.5, respectively. In the second period, the user firstly interacts with an adventure movie "Captain America". Thus, our method recommends another famous adventure movie "Fast & Furious" with 0.3. It is mainly because the user hasn't interacted with this kind of movie before. Followed by this, the user interacts with ”Iron Man”, which is directed by “Jon Favreau”. Hence, our method gives another “Jon Favreau” directed movie ”Iron Man 2” with 0.6. The higher probability is primarily because that the user already interacted with a movie directed “Jon Favreau” in the first period and the later one plays a positive strengthening effect for the recommendation. In the third period, the user interacts with another adventure movie “Spired Man” directed also by “Jon Favreau”. On account of both two features that have been existed in the former two-period time, our method reasonable guesses that the user would like a movie meeting these two features jointly. Therefore, “Iron Man 3” is recommended with 0.8.

6 Related Work

Generally, the related works in this paper can be grouped into four categories: sequential recommendation, recommendation with KGs, recommendation with RL, sequential explainable recommendation.

6.1 Sequential Recommendation

Sequential recommendations have been becoming a hot topic in recent years. Some pioneer sequential models, such as LSTM [10], RNN[5], and GRU [34, 42], etc, have made an outstanding performance in the sequential recommendation. These methods generally predict users' subsequent top-N recommendation items based on their previous behaviors and contextual information. For instance, [44] adopts a Tree-LSTM model to improve the representation by combining the syntactic structure and the semantic information, which achieves significantly better results than standard LSTM. Considering the cold start problem due to the insufficiency of users' feedback, Qiang et al. [6] propose a multi-view RNN model to dynamically learn the comprehensive item representation with latent, visual, and textual features for a further sequential recommendation. However, the monotonic temporal dependency of RNN in [6] impairs the users' short-term interest. To solve this problem, a hierarchical contextual attention-based GRU network [17] comprehensively exploits users' several current hidden states and contextual hidden state information to reflect their real interests. In addition, there are some other methods [13,15, 36] for sequentially embed users' historical behaviors. For example, Tang et al. [23] propose a convolutional sequence embedding recommendation model as a solution to address this problem. This model uses convolution filters to embed a sequence of users' items into an ”image” feature to capture the users' general preference and sequential patterns. However, the main drawback of the sequential recommendation method is the one-sided observation, which means it only can capture the features from a users' perspective. Nevertheless, the complicated and enormous relations between items also imply abundant information.

6.2 Recommendation with KGs

To deal with the upper problems, the KGs-based recommendations have been attracting substantial interest in the research community. These methods can be primarily divided into two groups: KGs-based embedding methods and path-based recommendation.

KGs-embedding based models [7,8,39] usually leverage KGs embedding techniques to guide the representation learning of users and items. For instance, to integrate large-scale structured and unstructured data of KGs, a KGs-based explainable collaborative filtering framework [1] is proposed, which utilizes a knowledge-base representation learning framework to embed heterogeneous entities and a soft matching algorithm to generate personalized explanations for the recommended items. Current collaborative filtering usually suffers from a poor recommendation performance due to the sparsity of user-item interaction. To address this problem, a collaborative knowledge base embedding framework [37] uses TransR to extract items' heterogeneous structural representations, which also applies stacked denoising auto-encoders and stacked convolutional auto-encoders to extract items' textual representations and visual representations, respectively. The KGs-embedding based models are flexible to exploit abundant embedding information from KGs. However, they lack an explicit explanation of relations in KGs for the final recommendation.

Different from KGs-embedding based models, the path-based recommendation usually explores the diverse relations among KGs to give an explicit and reliable explanation. For instance, a knowledge graph attention network [29] is proposed to exploit the higher-order reasoning paths, which recursively propagates the embeddings from a node's neighbors to refine the node's embedding and employ an attention mechanism to discriminate the importance of the neighbors. To further exploit the information encoded in KGs, [28] proposes an MRP2Rec to explore various semantic relations in multiple-step relation paths to improve recommendation performance. The above methods only consider relationships of as a single type. However, the recommendation problems in many applications exist in an attribute-rich heterogeneous network environment. To address this problem, a meta-path-based method [35] systematically learns the heterogeneous features to represent the different sizes of relationships between entities. Besides, Junwei et al. [38] use an attention-based bidirectional LSTM to learn the influence of different paths. The path-based recommendation methods can achieve superior recommendation performance as well as path-based reasoning. However, they are prone to generate redundancy information since they enumerate all possible paths.

6.3 Recommendation with RL

RL has been achieving remarkable performance in non- files such as Question Answering (QA) [2], music recommendation [31], demonstrating its excellent ability in understanding the environment. In recent years, to promoting the recommendation models to search meaningful paths rather than enumerate all possible paths in KGs, RL has been gradually introduced in recommendations. Some RL-based recommendation models [4, 9, 41] have achieved outstanding performance in recommendation. For example, Song et al. [20] proposed a knowledge-aware recommendation model to generates meaningful paths from users to relevant items by learning a walking policy on the user-item-entity graph, which is designed to deal with the data sparsity and cold start problems. Besides, a PGPR model [33] is also proposed, which can provide the recommendation system with an ability to simultaneously generating reasoning paths and accurate recommendations. Specifically, it contains a multi-hop function for calculating the relevancy between users and terminal items in one path, an innovative soft reward strategy for evaluating the effect of users' choices, and a user-conditional action pruning strategy to guide the model for searching efficiently and effectively paths in KGs. Above all, the RL-based recommendation method can endow the recommendation system with an excellent path reasoning process.

6.4 Sequential Explainable Recommendation

Recently, some research [12, 24, 27] have conducted sequential recommendations based on KGs and user-item interactions. For instance, Baocheng et al. [24] use a hybrid of graph neural network and a key-value memory network to extract users' sequential interest and semantic-based preference, which improves the strategy for constructing session graphs from interaction sequences for the sequential recommendation task. To solve the user-commodity sparseness in , a knowledge-guided reinforcement learning model is proposed, which designs a composite reward function to compute both sequence and knowledge level rewards. However, these methods cannot provide explanations of why these items are recommended to users. To address this problem, a novel explainable interaction-driven user modeling algorithm [12] employs multi-modal fusion to learn the importance scores for specific user-item pairs, which aims to capture the users' interaction-level dynamic preference. To achieve better sequential explainable recommendations, several studies explore users' potential interests comprehensively considering users' sequential historical behaviors and KGs. To better model the sequential dependencies within a path, Wang et al. [30] contribute a knowledge-aware path recurrent network to leverage the sequential relations within one path based on a newly designed weighted pooling operation. To better explore the effect of users' sequence and KGs on recommendation, a knowledge-aware reasoning network [47] not only develops an attention-based RNN to capture users' historical interests but adopts a hierarchical attention neural network to reason on paths. Although the above methods can achieve good performance in the sequential explainable recommendation, none of these have considered the KGs-based recommendation as a sequential problem.

7 Conclusion and Future work

This paper proposes an RSL-GRU architecture for the KGs-based sequential explainable recommendation. It explicitly exploits abundant information in users' historical behaviors associated with their KGs. Specifically, RSL-GRU uses the blocking strategy to build a sequential KGs. Besides, an RPRN is also designed for reasoning out the motivations behind each successful purchase behavior. To output potential top-N items for each user with appropriate reasoning paths from a global perspective, a GRU network combined with attention mechanism is utilized. We conduct the experiments on four Amazon e-commerce datasets to verify the excellent performance in both sequential recommendation and path reasoning compared with several state-of-art baselines. For future work, we would like to examine the RSL-GRU model on different recommendation tasks. We also intend to explore the heterogeneous information and contextual information of the paths in the future.

References

1. Ai, Q., Azizi, V., Chen, X., Zhang, Y.: Learning heterogeneous knowledge base embeddings for explainable recommendation. Algorithms **11**(9), 137 (2018)
2. Alkawsi, G.A., Ali, N., Baashar, Y.: An empirical study of the acceptance of iot-based smart meter in malaysia: The effect of electricity-saving knowledge and environmental awareness. IEEE Access **8**, 42794–42804 (2020)
3. Bianchi, F., Rossiello, G., Costabello, L., Palmonari, M., Minervini, P.: Knowledge graphs for explainable artificial intelligence: Foundations, applications and challenges. In: Studies on the Semantic Web, vol. 47, pp. 49–72. IOS Press (2020)
4. Chen, X., Huang, C., Yao, L., Wang, X., Liu, W., Zhang, W.: Knowledge-guided deep reinforcement learning for interactive recommendation. In: 2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020, pp. 1–8 (2020)
5. Cui, Q., Wu, S., Liu, Q., Zhong, W., Wang, L.: Mv-rnn: A multi-view recurrent neural network for sequential recommendation. IEEE Trans. Knowl. Data Eng. **32**(2), 317–331 (2016)
6. Cui, Q., Wu, S., Liu, Q., Zhong, W., Wang, L.: MV-RNN: A multi-view recurrent neural network for sequential recommendation. IEEE Trans. Knowl. Data Eng. **32**(2), 317–331 (2020)
7. Dai, F., Gu, X., Li, B., Zhang, J., Qian, M., Wang, W.: Meta-graph based attention-aware recommendation over heterogeneous information networks. In: Computational Science - ICCS 2019 - 19th International Conference, Faro, Portugal, June 12-14, 2019, Proceedings, Part II, pp. 580–594 (2019)
8. Gong, J., Wang, S., Wang, J., Feng, W., Peng, H., Tang, J., Yu, P.S.: Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view. In: Proceedings of the 43rd International conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020, pp. 79–88 (2020)
9. He, X., An, B., Li, Y., Chen, H., Wang, R., Wang, X., Yu, R., Li, X., Wang, Z.: Learning to collaborate in multi-module recommendation via multi-agent reinforcement learning without communication. In: RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020, pp. 210–219 (2020)
10. Heinz, S., Bracher, C., Vollgraf, R.: An lstm-based dynamic customer model for fashion recommendation. CEUR-WS.org **1922**, 45–49 (2017)

11. Hu, B., Shi, C., Zhao, W.X., Yu, P.S.: Leveraging meta-path based context for top- N recommendation with A neural co-attention model. In: Proceedings of the 24th ACM SIGKDD International Conference Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018, pp. 1531–1540 (2018)
12. Huang, X., Fang, Q., Qian, S., Sang, J., Li, Y., Xu, C.: Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In: Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019, pp. 548–556 (2019)
13. Kolahkaj, M., Harounabadi, A., Nikravanshalmani, A., Chinipardaz, R.: A hybrid context-aware approach for e-tourism package recommendation based on asymmetric similarity measurement and sequential pattern mining. Electron. Commer. Res. Appl. **42**, 100978 (2020)
14. Li, L., Wang, P., Wang, Y., Jiang, J., Tang, B., Yan, J., Wang, S., Liu, Y.: Prtransh: Embedding probabilistic medical knowledge from real world emr data. CoRR **abs/1909.00672**(8) (2019)
15. Ma, C., Ma, L., Zhang, Y., Sun, J., Liu, X., Coates, M.: Memory augmented graph neural networks for sequential recommendation. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020, pp. 5045–5052 (2020)
16. Palumbo, E., Rizzo, G., Troncy, R., Baralis, E., Osella, M., Ferro, E.: Knowledge graph embeddings with node2vec for item recommendation. In: The Semantic Web: ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers, *Lecture Notes in Computer Science*, vol. 11155, pp. 117–120. Springer (2018)
17. Qiang, C., Shu, W., Yan, H., Liang, W.: A hierarchical contextual attention-based gru network for sequential recommendation. Neurocomputing **abs/1711.05114**(8) (2017)
18. Saito, T., Watanobe, Y.: Learning path recommendation system for programming education based on neural networks. IJDET **18**(1), 36–64 (2020)
19. Shi, D., Wang, T., Xing, H., Xu, H.: A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning. Knowl. Based Syst. **195**, 105618 (2020)
20. Song, W., Duan, Z., Yang, Z., Zhu, H., Zhang, M., Tang, J.: Explainable knowledge graph-based recommendation via deep reinforcement learning. CoRR **abs/1906.09506**, 13 (2019)
21. Sun, Z., Huang, J., Hu, W., Chen, M., Guo, L., Qu, Y.: Transedge: Translating relation-contextualized embeddings for knowledge graphs. CoRR **abs/2004.13579**(17) (2020)
22. Suzuki, T., Oyama, S., Kurihara, M.: Explainable recommendation using review text and a knowledge graph. In: 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019, pp. 4638–4643. IEEE (2019)
23. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018, pp. 565–573. ACM (2018)
24. Wang, B., Cai, W.: Knowledge-enhanced graph neural networks for sequential recommendation. Inf. **11**(8), 388 (2020)
25. Wang, L., Wang, Y., Liu, B., He, L., Liu, S., de Melo, G., Xu, Z.: Link prediction by exploiting network formation games in exchangeable graphs. In: 2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017, pp. 619–626. IEEE (2017)
26. Wang, L., Zhang, W., He, X., Zha, H.: Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018, pp. 2447–2456. ACM (2018)
27. Wang, P., Fan, Y., Xia, L., Zhao, W.X., Niu, S., Huang, J.: 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020, pp. 209–218 (2020)
28. Wang, T., Shi, D., Wang, Z., Xu, S., Xu, H.: Mrp2rec: Exploring multiple-step relation path semantics for knowledge graph-based recommendations. IEEE Access **8**, 134817–134825 (2020)

29. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S.: Kgat: Knowledge graph attention network for recommendation. ACM **9**(9), 950–958 (2019)
30. Wang, X., Wang, D., Xu, C., He, X., Cao, Y., Chua, T.: Explainable reasoning over knowledge graphs for recommendation. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, 8, pp. 5329–5336. AAAI Press (2019)
31. Wang, X., Wang, Y., Hsu, D., Wang, Y.: Exploration in interactive personalized music recommendation: A reinforcement learning approach. Acm Transactions on Multimedia Computing Communications Applications **11**(1), 1–22 (2013)
32. Wang, Z., Li, Y., Fang, L., Chen, P.: Joint knowledge graph and user preference for explainable recommendation. In: 2019 IEEE 5th International Conference on Computer and Communications (ICCC), pp. 1338–1342. IEEE (2019)
33. Xian, Y., Fu, Z., Muthukrishnan, S., de Melo, G., Zhang, Y.: Reinforcement knowledge graph reasoning for explainable recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, pp. 285–294. ACM (2019)
34. Yang, C., Sun, M., Zhao, W.X., Liu, Z., Chang, E.Y.: A neural network approach to joint modeling social networks and mobile trajectories. Acm Transactions on Information Systems **35**(4), 36 (2016)
35. Yu, X., Ren, X., Sun, Y., Sturt, B., Khandelwal, U., Gu, Q., Norick, B., Han, J.: Recommendation in heterogeneous information networks with implicit user feedback. In: Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013, pp. 347–350 (2013)
36. Yuan, W., Wang, H., Yu, X., Liu, N., Li, Z.: Attention-based context-aware sequential recommendation model. Inf. Sci. **510**, 122–134 (2020)
37. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pp. 353–362. ACM (2016)
38. Zhang, J., Gao, M., Yu, J., Yang, L., Wang, Z., Xiong, Q.: Path-based reasoning over heterogeneous networks for recommendation via bidirectional modeling. CoRR **abs/2008.04185** (2020)
39. Zhao, B., Xu, Z., Tang, Y., Li, J., Liu, B., Tian, H.: Effective knowledge-aware recommendation via graph convolutional networks. In: Web Information Systems and Applications - 17th International Conference, WISA 2020, Guangzhou, China, September 23-25, 2020, Proceedings, pp. 96–107 (2020)
40. Zhao, H., Yao, Q., Li, J., Song, Y., Lee, D.L.: Meta-graph based recommendation fusion over heterogeneous information networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13- 17, 2017, pp. 635–644. ACM (2017)
41. Zhao, K., Wang, X., Zhang, Y., Zhao, L., Liu, Z., Xing, C., Xie, X.: Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020, pp. 239–248 (2020)
42. ZHAO, Z., ZHU, M., SHENG, Y., WANG, J.: A top-n-balanced sequential recommendation based on recurrent network. Ieice Transactions on Information Systems **102**(4), 737–744 (2019)
43. Zheng, L., Tianlong, Z., Huijian, H., Caiming, Z.: Personalized tag recommendation based on convolution feature and weighted random walk. International Journal of Computational Intelligence Systems **13**(1), 24–35 (2020)
44. Zhu, Rui, Yang, Delu, Li: Learning improved semantic representations with tree-structured lstm for hashtag recommendation: An experimental study. Information **10**(4), 127 (2019)
45. Zhu, H., Tian, F., Wu, K., Shah, N., Chen, Y., Ni, Y., Zhang, X., Chao, K.M., Zheng, Q.: A multi-constraint learning path recommendation algorithm based on knowledge map. Knowledge Based Systems **143**(MAR.1), 102–114 (2018)
46. Zhu, Q., Zhou, X., Song, Z., Tan, J., Guo, L.: DAN: deep attention neural network for news recommendation. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, pp. 5973–5980. AAAI Press (2019)

47. Zhu, Q., Zhou, X., Wu, J., andLi Guo, J.T.: A knowledge-aware attentional reasoning network for recommendation. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, 8, pp. 6999–7006. AAAI Press (2020)