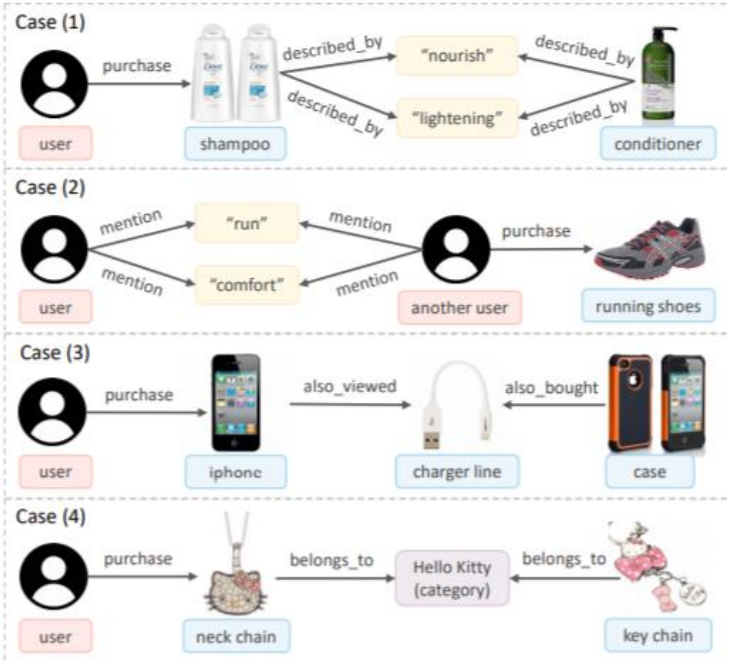
Reinforcement Knowledge Graph Reasoning for Explainable Recommendation

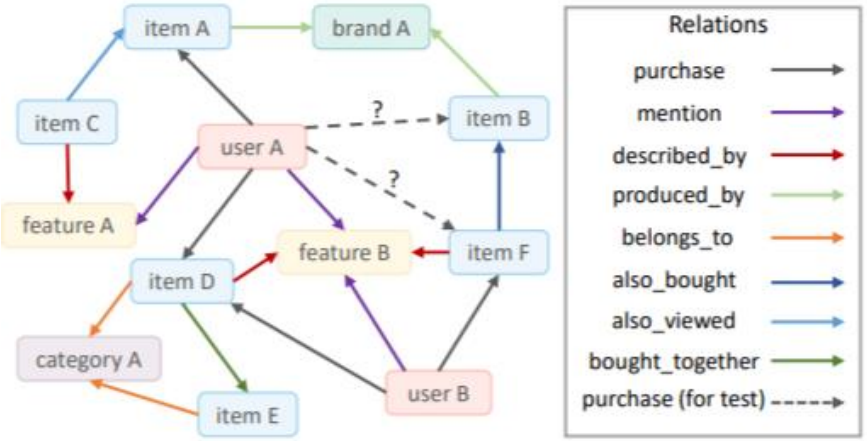
SIGIR 2019

Research Topic

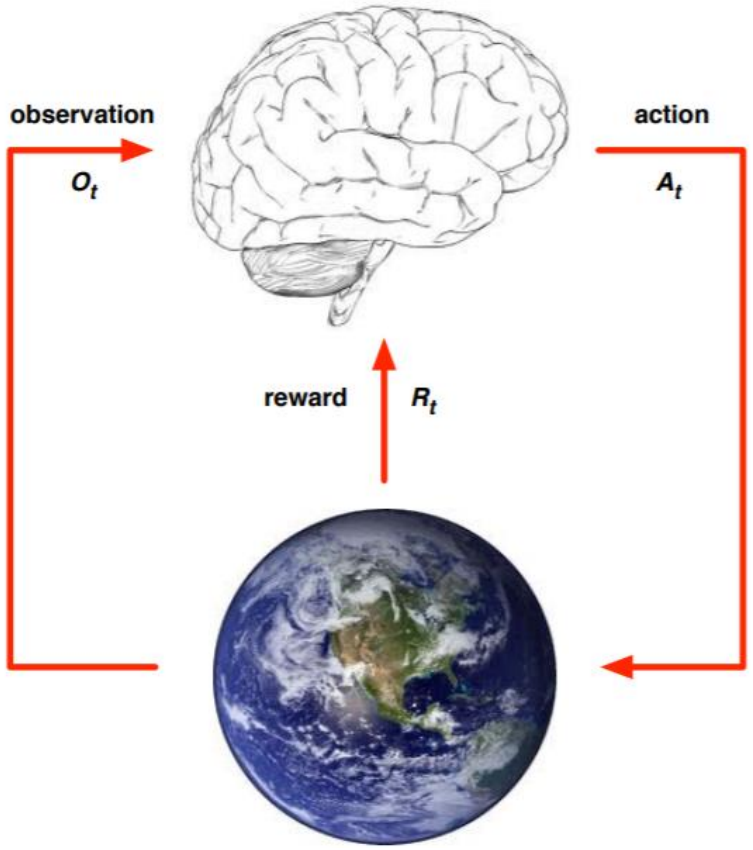


Entities	Description
User	User in recommender system
Item	Product to be recommended to users
Feature	A product feature word from reviews
Brand	Brand or manufacturer of the product
Category	Category of the product

Relations	Description
Purchase	*User* $\xrightarrow{\text{purchase}}$ *Item*
Mention	*User* $\xrightarrow{\text{mention}}$ *Feature*
Described_by	*Item* $\xrightarrow{\text{described\_by}}$ *Feature*
Belong_to	*Item* $\xrightarrow{\text{belong\_to}}$ *Category*
Produced_by	*Item* $\xrightarrow{\text{produced\_by}}$ *Brand*
Also_bought	*Item* $\xrightarrow{\text{also\_bought}}$ *Item*
Also_viewed	*Item* $\xrightarrow{\text{also\_viewed}}$ another *Item*
Bought_together	*Item* $\xrightarrow{\text{bought\_together}}$ another *Item*



Research Topic



Deterministic Markov Decision Process

an agent starts from a given user, learns to navigate to the potential items of interest, such that the path history can serve as a genuine explanation for why the item is recommended to the user.

Related Work

- knowledge graph embedding model(eg: Collaborative Filtering)

1)align the knowledge graph in a regularized vector space

2)uncover the similarity between entities by calculating their representation distance

find a post-hoc explanation for the already chosen recommendations

- path-based recommendation

enumerates all the qualified paths between every user–item pair, and then trained a sequential RNN model from the extracted paths to predict the ranking score

Advantage

(1) We highlight the significance of incorporating rich heterogeneous information into the recommendation problem to formally define and interpret the reasoning process.

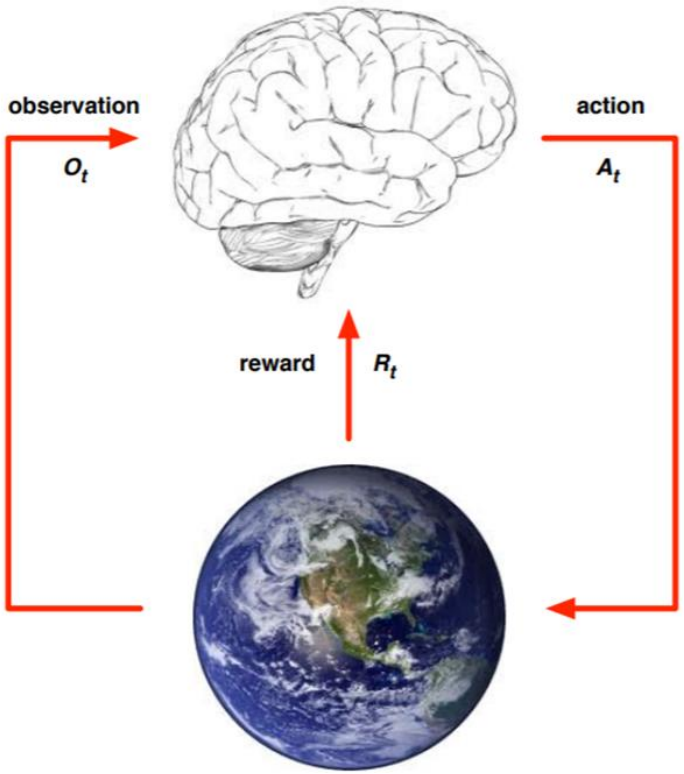
(2) We propose an RL-based approach to solve the problem,driven by our soft reward strategy, user-conditional action pruning, and a multi-hop scoring strategy.

(3) We design a beam search-based algorithm guided by the policy network to efficiently sample diverse reasoning paths and candidate item sets for recommendation.

(4) We extensively evaluate the effectiveness of our method on several Amazon e-commerce domains, obtaining strong results as well as explainable reasoning paths.

Methodology

Markov Decision Process



- State. The state st at step t is defined as a tuple (u, et ,ht)

  et is the entity the agent has reached at step t, and ht is the history prior to step t.

- Action. $\tilde{A}_t(u) = \{(r,e) \mid \text{rank}(f((r,e) \mid u)) \leq \alpha, (r,e) \in A_t\}$, user-conditional action pruning strategy

- Reward. Given any user, there is no pre-known targeted

  a soft reward $R_T = \begin{cases} \max\left(0, \frac{f(u,e_T)}{\max_{i \in \mathcal{I}} f(u,i)}\right), & \text{if } e_T \in \mathcal{I} \\ 0, & \text{otherwise,} \end{cases}$

- Optimization. learn a stochastic policy π that maximizes the expected cumulative reward for any initial user u:

  policy network value network

  $\nabla_\Theta J(\Theta) = \mathbb{E}_\pi \left[\nabla_\Theta \log \pi_\Theta(\cdot|\mathbf{s}, \tilde{\mathbf{A}}_u)\,(G - \hat{v}(\mathbf{s}))\right]$

Methodology

Policy-Guided Path Reasoning

- This method cannot guarantee the diversity of paths, because the agent guided by the policy network is likely to repeatedly search the same path with the largest cumulative rewards.
- By the action probability and reward to explore the candidate paths as well as the recommended items for each user.

Algorithm 1: Policy-Guided Path Reasoning

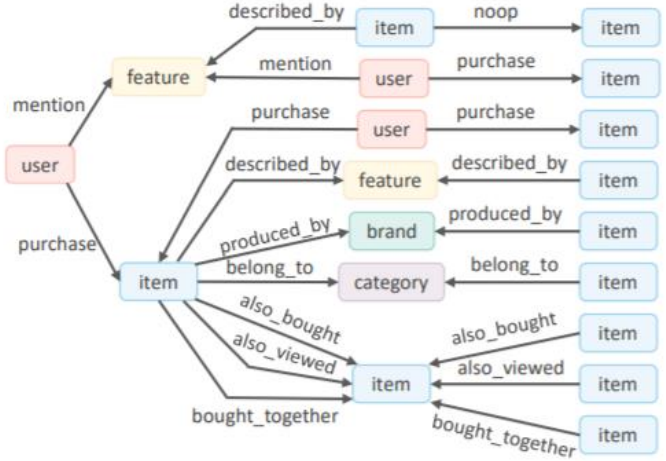
Input : $u, \pi(\cdot|\mathbf{s}, \tilde{\mathbf{A}}_u), T, \{K_1, \ldots, K_T\}$
Output: path set $\mathcal{P}_T$, probability set $\mathcal{Q}_T$, reward set $\mathcal{R}_T$

Initialize $\mathcal{P}_0 \leftarrow \{\{u\}\}, \mathcal{Q}_0 \leftarrow \{1\}, \mathcal{R}_0 \leftarrow \{0\}$;
for $t \leftarrow 1$ **to** T **do**
  Initialize $\mathcal{P}_t \leftarrow \emptyset, \mathcal{Q}_t \leftarrow \emptyset, \mathcal{R}_t \leftarrow \emptyset$;
  forall $\hat{p} \in \mathcal{P}_{t-1}, \hat{q} \in \mathcal{Q}_{t-1}, \hat{r} \in \mathcal{R}_{t-1}$ **do**
    $\triangleright$ path $\hat{p} \doteq \{u, r_1, \ldots, r_{t-1}, e_{t-1}\}$;
    Set $s_{t-1} \leftarrow (u, e_{t-1}, h_{t-1})$;
    Get user-conditional pruned action space $\tilde{A}_{t-1}(u)$ from environment given state s_{t-1};
    $\triangleright$ $p(a) \doteq \pi(a \mid \mathbf{s}_{t-1}, \tilde{\mathbf{A}}_{u,t-1})$ and $a \doteq (r_t, e_t)$;
    Actions $\mathcal{A}_t \leftarrow \{a \mid \text{rank}(p(a)) \leq K_t, \ \forall a \in \tilde{A}_{t-1}(u)\}$;
    forall $a \in \mathcal{A}_t$ **do**
      Get s_t, R_{t+1} from environment given action a;
      Save new path $\hat{p} \cup \{r_t, e_t\}$ to $\mathcal{P}_t$;
      Save new probability $p(a)\,\hat{q}$ to $\mathcal{Q}_t$;
      Save new reward $R_{t+1} + \hat{r}$ to $\mathcal{R}_t$;
    end
  end
end
Save $\forall \hat{p} \in \mathcal{P}_T$ if the path $\hat{p}$ ends with an item;
return filtered $\mathcal{P}_T, \mathcal{Q}_T, \mathcal{R}_T$;

https://blog.csdn.net/qq_38871942

Experiment

Dataset	CDs & Vinyl				Clothing				Cell Phones				Beauty			
Measures (%)	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.
BPR	2.009	2.679	8.554	1.085	0.601	1.046	1.767	0.185	1.998	3.258	5.273	0.595	2.753	4.241	8.241	1.143
BPR-HFT	2.661	3.570	9.926	1.268	1.067	1.819	2.872	0.297	3.151	5.307	8.125	0.860	2.934	4.459	8.268	1.132
VBPR	0.631	0.845	2.930	0.328	0.560	0.968	1.557	0.166	1.797	3.489	5.002	0.507	1.901	2.786	5.961	0.902
TransRec	3.372	5.283	11.956	1.837	1.245	2.078	3.116	0.312	3.361	6.279	8.725	0.962	3.218	4.853	0.867	1.285
DeepCoNN	4.218	6.001	13.857	1.681	1.310	2.332	3.286	0.229	3.636	6.353	9.913	0.999	3.359	5.429	9.807	1.200
CKE	4.620	6.483	14.541	1.779	1.502	2.509	4.275	0.388	3.995	7.005	10.809	1.070	3.717	5.938	11.043	1.371
JRL	5.378*	7.545*	16.774*	2.085*	1.735*	2.989*	4.634*	0.442*	4.364*	7.510*	10.940*	1.096*	4.396*	6.949*	12.776*	1.546*
PGPR (Ours)	**5.590**	**7.569**	**16.886**	**2.157**	**2.858**	**4.834**	**7.020**	**0.728**	**5.042**	**8.416**	**11.904**	**1.274**	**5.449**	**8.324**	**14.401**	**1.707**



Thank you for listening!