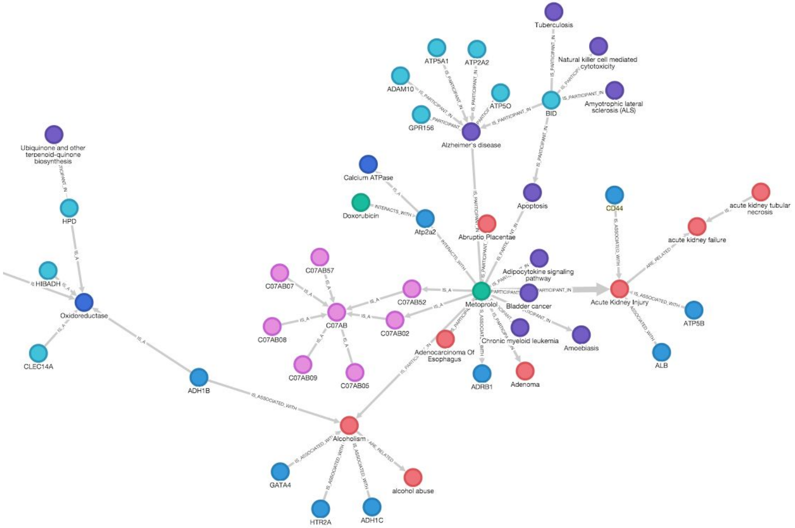
Reinforcement Knowledge Graph Reasoning for Explainable Recommendation

David Dimmett, William Helmrath, Ram Karri, Jihong Min

Introduction

- Personal recommendation has sparked a lot of interest in rich structured information in the form of knowledge graphs
- Most existing approaches only focus on using knowledge graphs for more accurate recommendation



Introduction

- This paper performs explicit reasoning with knowledge for decision making so that the recommendations are supported by an interpretable causal inference procedure.
- They propose a method called Policy-Guided Path Reasoning, which uses both recommendation and interpretability by providing actual paths in a knowledge path.

Introduction

4 main contributions

- Highlight the significance of incorporating knowledge graphs into recommendation to formally define and interpret the reasoning process
- Propose a reinforcement learning (RL) approach featuring an innovative soft reward strategy, user-conditional action pruning and a multi-hop scoring function
- Design a policy-guided graph search algorithm to efficiently and effectively sample reasoning paths for recommendation
- Extensively evaluate our method on several large-scale real-world benchmark datasets, obtaining favorable results compared with state-of-the-art methods

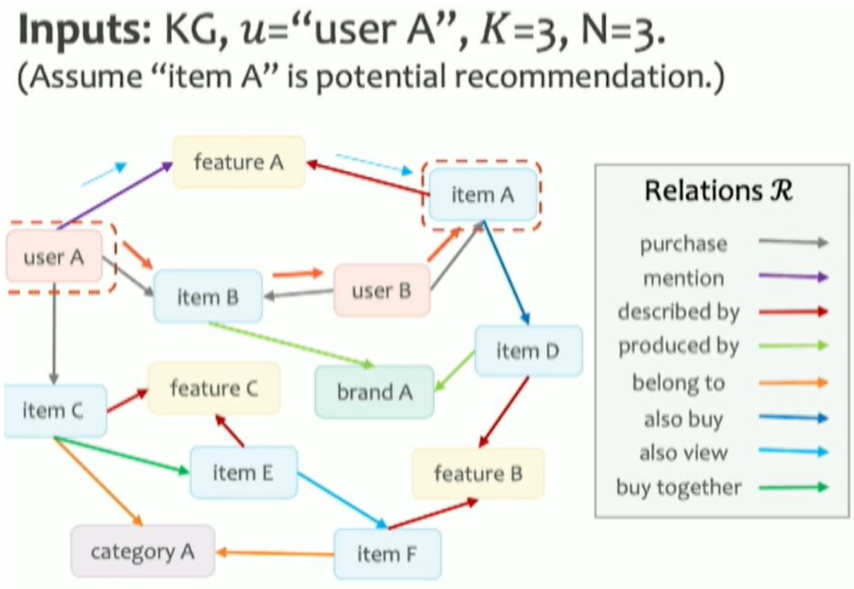
Methodology: Problem Formulation 1

KGRE-Rec Problem

Definition 3.2. (KGRE-Rec Problem) Given a knowledge graph $\mathcal{G}_{\mathrm{R}}$, user $u \in \mathcal{U}$ and integers K and N, the goal is to find a recommendation set of items $\{i_n\}_{n \in [N]} \subseteq \mathcal{I}$ such that each pair (u, i_n) is associated with one reasoning path $p_k(u, i_n)$ $(2 \leq k \leq K)$, and N is the number of recommendations.

Inputs: knowledge graph, user
Outputs: set of recommended items such that each pair of user and set of items is associated with one reasoning path

Example



Algorithm would output one of the following paths:

- "user A -> "purchase" -> "item B" -> "purchase" -> "user B" -> "purchase" -> "item A"
- "user A" -> "mention" -> "feature A" -> "described_by" -> "item A"

Methodology: Problem Formulation 2

Goal: Simultaneously conduct item recommendation and path finding

Challenges:

- Do not have pre-defined targeted items for any user
- Not applicable to use a binary reward indicating whether the user interacts with the item or not
- Out-degrees of some entities may be very large, degrading efficiency of finding paths

Important Aspects

- Incorporate the uncertainty of how an item is relevant to a user based on the **rich heterogeneous information** given by the knowledge graph (KG)
- Effectively perform edge pruning and efficiently search relevant paths towards potential items using the reward as a heuristic
- Diversity of reasoning paths for recommended items should be guaranteed for every user

Overall Solution

Policy-Guided Path Reasoning (PGPR) method for explainable recommendation over knowledge graphs

- Solves the problem through reinforcement learning (RL)
  - making recommendations while simultaneously searching paths in the context of rich heterogeneous information in the knowledge graph
- Train an RL agent that learns to navigate to potentially "good" items conditioned on the starting user in the KG environment
- Agent is then exploited to efficiently sample reasoning paths for each user leading to the recommended items
- Sampled paths serves as the explanations for the recommended items

RL-based Approach

User-Conditional Action Pruning Strategy

- Keeps the promising edges conditioned on the starting user based on a scoring function.
- Scoring function maps any edge to a real-valued score conditioned on a user

Soft Reward Strategy

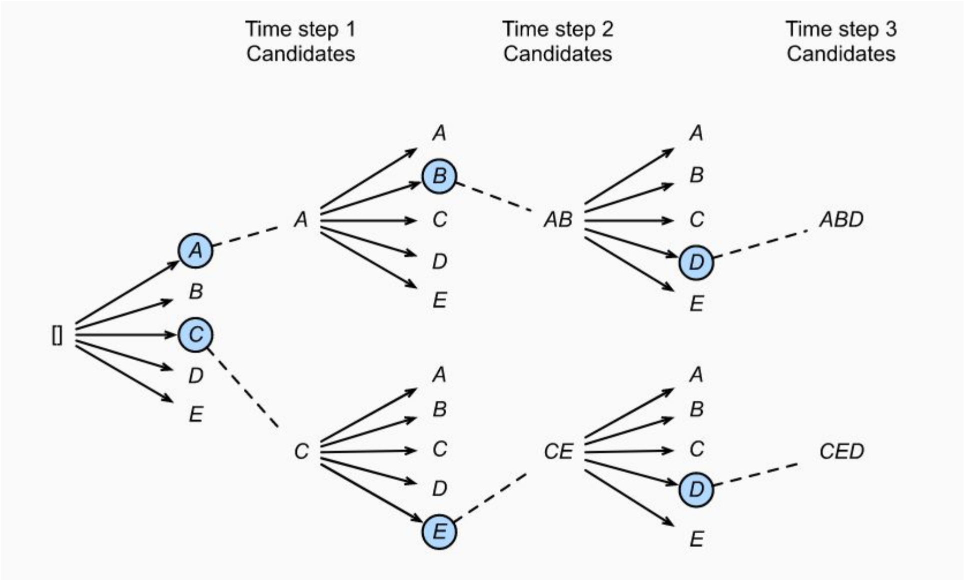
- Unfeasible to consider binary rewards indicating whether the agent has reached a target or not
- Instead agent is encouraged to explore as many “good” paths as possible. A “good path” is one that leads to an item that a user will interact with.

Methodology: Beam Search-Based Algorithm

- Most search methods will return similar reasoning paths from a user to candidate items
- Solution: use **beam search** to return a wide variety of recommended items, ranked based on their path reward
- In the case that multiple sequences result in the same item $\boldsymbol{i_n}$, choose the path with the highest generative probability

Methodology: Beam Search-Based Algorithm

Beam search is a greedy breadth-first search algorithm that chooses the *k* tokens with the highest conditional probability at each time step. *k* is a predetermined parameter called the “beam size”. At each time step, the current sequence is returned as a possible output sequence



k = 2

Experimentation: Data Description

- Data from 4 categories on Amazon: CDs and Vinyl, Clothing, Cell Phones, and Beauty
- Each category was treated separately during the experiment
- Data was preprocessed using a frequency metric that removed irrelevant words
- 70% of data was used for training, 30% for testing

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

Experiment Setup: Other Models

- Models used for comparison:
    - BPR - Bayesian personalized ranking model
    - BPR-HFT - Hidden Factors and Topics model
    - VBPR - Visual Bayesian Personalized Ranking
    - TransRec
    - DeepCoNN - Deep Cooperative Neural Networks
    - CKE - Collaborative Knowledge base Embedding
    - JRL

Experiment Setup: Evaluation Methods

- Each model was applied to each category and evaluated using:
  - NDCG
  - Recall
  - HR
  - Precision
- Metrics were based on top 10 predictions

Experiment Results

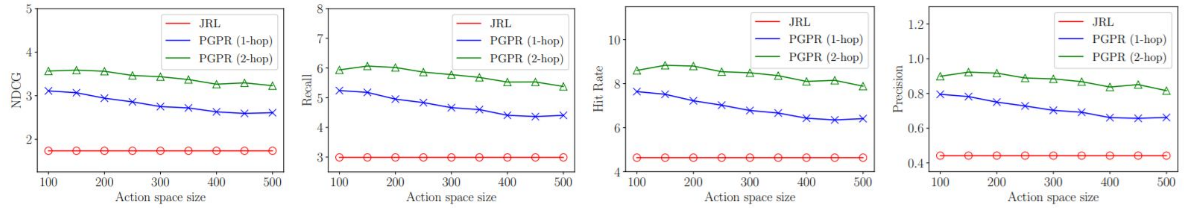
- Results were compelling: PGPR outperformed all other methods in every metric across the 4 categories
- Success rate for finding valid paths: 50%
- PGPR was able to find multiple lines of reasoning for some recommendations
- Performance improved with smaller action space → good scoring function

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**

Table 2: Overall recommendation effectiveness of our method compared to other baselines on four Amazon datasets. The results are reported in percentage (%) and are calculated based on the top-10 predictions in the test set. The best results are highlighted in bold and the best baseline results are marked with a star (∗).

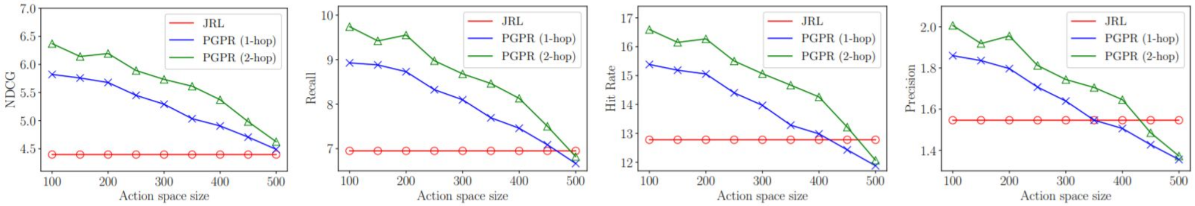
Experiment: Multi-Hop Scoring Function

Determine whether multi-hop scoring functions can improve recommendation performance as opposed to the default 1-hop



(a) NDCG (b) Recall (c) Hit Rate (d) Precision

Figure 3: Recommendation effectiveness of our model under different sizes of pruned action spaces on the *Clothing* dataset. The results using multi-hop scoring function are also reported.



(a) NDCG (b) Recall (c) Hit Rate (d) Precision

Figure 4: Recommendation effectiveness of our model under different sizes of pruned action spaces on the *Beauty* dataset. The results using multi-hop scoring function are also reported.

Experiment: Sampling Size in Path Reasoning

Determine how the sampling size for path reasoning influences recommendation performance

Dataset	Clothing				Beauty			
Sizes	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.
25, 5, 1	<u>2.858</u>	<u>4.834</u>	<u>7.020</u>	<u>0.728</u>	<u>5.449</u>	<u>8.324</u>	<u>14.401</u>	<u>1.707</u>
20, 6, 1	2.918	4.943	7.217	0.749	5.555	8.470	14.611	1.749
20, 3, 2	2.538	4.230	6.177	0.636	4.596	6.773	12.130	1.381
15, 8, 1	2.988	5.074	7.352	0.767	5.749	8.882	15.268	1.848
15, 4, 2	2.605	4.348	6.354	0.654	4.829	7.138	12.687	1.458
12, 10, 1	3.051	5.207	7.591	0.791	5.863	9.108	15.599	1.905
12, 5, 2	2.700	4.525	6.575	0.679	4.968	7.365	13.168	1.519
10, 12, 1	**3.081**	**5.271**	**7.673**	**0.797**	**5.926**	**9.166**	**15.667**	**1.920**
10, 6, 2	2.728	4.583	6.733	0.693	5.067	7.554	13.423	1.559

Table 4: Influence of sampling sizes at each level on the recommendation quality. The best results are highlighted in bold and the results under the default setting are underlined. All numbers in the table are given in percentage (%).

Experiment: History Representations

Determine how different representations of state history influence the method

Dataset	Clothing				Beauty			
History	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.
0-step	1.972	3.117	4.492	0.462	3.236	4.407	8.026	0.888
1-step	2.858	4.834	7.020	0.728	5.449	8.324	14.401	1.707
2-step	2.786	4.702	6.865	0.710	5.342	8.181	14.168	1.669

Table 5: Results for different history representations of state. All numbers in the table are given in percentage (%).

Conclusions and Future Work

- Future models should have the ability to perform explicit reasoning over knowledge for decision making
- For future work, PGPR can be used for many other graph based tasks including product search and social recommendation.
- PGPR can also be used to model time-evolving graphs for dynamic decision support.

Works Cited

1. Graphics: http://gerard.demelo.org/papers/kg-recommendations.pdf