Sequence-aware Reinforcement Learning over Knowledge Graphs

Ashish Gupta
Walmart Labs
ashish.gupta@walmartlabs.com

Rishabh Mehrotra
Spotify Research
rishabhm@spotify.com

ABSTRACT

We consider the task of generating explainable recommendations with knowledge graphs in a large scale industrial e-commerce platform. We propose a Reinforcement Learning (RL) based approach for recommendation, which casts item recommendation problem as a deterministic Markov Decision Process (MDP) over the knowledge graph, wherein an agent starts from a user, and learns to navigate to the potential items of interest. We hypothesize that the path history can serve as a genuine explanation for why the item is recommended to the user. Different from past work on RL on knowledge graphs [9], we leverage sequential neural modeling of user's historic item history, and hierarchical softmax approach for sampling paths in the knowledge graphs and propose Sequence Aware Reinforced Learning over Knowledge Graphs (SeqReLG). Experiments on large scale real world dataset highlights the benefits offered by sequential modeling of user's history and action sampling techniques. We observe a significant gain in performance when compared to state-of-the-art RL based approach. We additionally discuss and address implementation details for large scale deployment of the proposed RL based solution.

1 INTRODUCTION

Most recommendation settings suffer from a data sparsity issue wherein a large portion of products and items suffer from lack of interaction information. Knowledge graphs are increasingly being used to provide personalized recommendations to users in environments suffering from sparsity issues [1, 6, 9]. Knowledge graphs help encode and leverage heterogenous information, such as purchases, ratings, reviews and clicks in the modeling phase.

In this paper, we consider knowledge graphs as a versatile structure to maintain the agent's knowledge about users, items, other entities and their relationships. We build on top of recent work which leverages reinforcement learning over knowledge graphs [7] and propose a Sequence-aware Reinforced Learning model over Knowledge Graphs (SeqReLG). The agent starts from a user and conducts explicit multi-step path reasoning over the graph, so as to discover suitable items in the graph for recommendation to the target user. The underlying idea is that if the agent draws its conclusion based on an explicit reasoning path, it will be easy to interpret the reasoning process that leads to each recommendation. Thus, the system can provide causal evidence in support of the recommended items.

Accordingly, our goal is not only to select a set of candidate items for recommendation, but also to provide the corresponding reasoning paths in the graph as interpretable evidence for why a given recommendation is made. Unlike previous approaches to the problem [7], the proposed model encodes the sequential interaction history of the user in the RL framework and leverages hierarchical softmax technique to sample actions. Large scale experiments on a real world dataset of product purchases demonstrate the efficacy of the proposed approach on the item recommendation task. We present preliminary results comparing the proposed approach to a number of baselines and most recent state-of-the-art model. Our findings have implications on the design of sequence aware reinforcement learning models for recommendations over knowledge graphs.

, ,
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 SEQUENCE-AWARE RL OVER GRAPHS

We begin by describing the RL based approach proposed by Yikun *et al.* [7] for generating recommendations from the user-item knowledge graph. We next highlight major shortcomings of the proposed approach and present Sequence-aware Reinforced Learning over Knowledge Graphs (SeqReLG). Finally, we discuss empirical results which demonstrate the efficacy of the proposed approach.

2.1 Reinforcement Knowledge Graph Reasoning

Yikun *et al.* [7] pose the recommendation problem as that of finding a recommendation set of items such that each pair of user, item is associated with one reasoning path in a graph in a knowledge graph of users and items. A knowledge graph G with entity set E and relation set R is defined as $G = (e, r, e')|e, e' \in E, r \in R$, where each triplet (e,r,$e^{'}$) represents a fact of the relation r from head entity e to tail.

The state at step t is defined as a tuple (u, e_t, h_t), where $u \in U$ is the starting user entity, e_t is the entity the agent has reached at step t, and h_t is the history prior to step t. The complete action space A_t of state s_t is defined as all possible outgoing edges of entity et excluding history entities and relations. During recommendation stage, the agent is encouraged to explore as many *good* paths as possible, such that the path leads to an item that a user will interact with, with high probability. To this end, a soft reward is estimated for the terminal state $s_T = (u, e_T, h_T)$ based on a scoring function $f(u, i)$. Based on our MDP formulation, the goal is to learn a stochastic policy that maximizes the expected cumulative reward for any user u. The problem is solved through REINFORCE with baseline[5] by designing a policy network and a value network that share the same feature layers.

The final step is to solve our recommendation problem over the knowledge graph guided by the trained policy network. A beam search guided by the action probability and reward is employed to explore the candidate paths as well as the recommended items for each user. For each pair of (u, e) in the candidate set, the path

with the highest generative probability is selected, the selected interpretable paths are ranked according to the path reward and corresponding items are recommended to the user.

2.2 Sequence Aware RL over Graphs

The proposed SeqReLG employs sequential neural model to model history sequences of user states and hierarchical sequences to sample actions from the different possile recommendation paths in the knowledge graph. We next describe both these in detail.

2.2.1 Policy Network: In this paper, we use the policy gradient method to solve the proposed recommendation MDP. Based on parameterized state s_t and parameterized action a, we calculate the probability distribution over possible action A_t as follows:

$$s_t = BiLSTM(s_{t-1}, [r_t, e_t]) \forall t > 0 \tag{1}$$

$$y_t = W_2 ReLU(W_1 s_t + b_1) + b_2 \tag{2}$$

$$\pi_\theta(a^{'}|s_t) = softmax(a^{'T} y_t) \tag{3}$$

$$\hat{v}(s) = y_t * W_v \tag{4}$$

where W_1, W_2, W_v and b_1, b_2 are weight matrices and weight vectors of a three layer neural network and $\pi_\theta(a^{'}|s_t)$ is the probability of taking action $a^{'}$ under state s_t.

2.2.2 Optimization: Our goal is to learn a stochastic policy π that maximizes the expected cumulative reward for any initial user u:

$$J(\theta) = E_\pi[\sum_{t=0}^{T} \gamma^t R_{t+1} | s_0 = (u, u, \phi)] \tag{5}$$

We solve the problem through REINFORCE with baseline[5] by designing a policy network and a value network that share the same feature layers as discussed in above equations and hence, we wish to maximize J(θ) via gradient ascent i.e.

$$\nabla_\theta J(\theta) = E_\theta[\nabla_\theta log \pi_\theta(a'|s_t)(G - \hat{v}(s))] \tag{6}$$

where G is discounted cumulative reward from state s to terminal state s_T.

2.2.3 Action Dropout: Since the action space is huge and enumeration of all possible paths between each user and all items is unfeasible on very large graphs. Hence, we need to find meaningful paths and perform action dropout carefully so that relevant actions don't get dropped out. However, due to the huge size of the entity set E, we adopt a hierarchical softmax technique[4] to approximate the *scoring function* for action dropout. Our scoring function is how we define the rewards. Hence, we maximize:

$$P(e'|e, r) = softmax(f(e, e'|r)) \tag{7}$$

and then perform *hierarchical softmax* on it so as to maximize the objective function:

$$J(G) = \prod_{e, e' \in E} I(e, r, e') * P(e'|e, r) \tag{8}$$

where I(e,r,$e^{'}$) is 1 if r is valid relation between e and $e^{'}$ else 0 and E is the entity set consisting of user and items in the graph.

Method	NDCG	Recall	HR	Precision
BPR-HFT	1.067	1.819	2.872	0.297
VBPR	0.560	0.968	1.557	0.166
TransRec	1.245	2.078	3.116	0.312
DeepCoNN	1.310	2.332	3.286	0.229
CKE	1.502	2.509	4.275	0.388
JRL	1.735	2.989	4.634	0.442
PGPR [7]	2.858	4.834	7.020	0.728
SeqReLG	**2.934**	**5.021**	**7.212**	**0.749**

Table 1: Performance of different methods.

3 EXPERIMENTS

We perform preliminary experiments on data obtained from a real world e-commerce website. The dataset consists of a random sample of click and purchase history of over 1 million users and 500K products over a span of 3 weeks. We perform a hold-out style evaluation on historic log data, and compare a number of established baselines in addition to recent state-of-the-art approaches.

BPRHFT[3] is a Hidden Factors and Topics (HFT) model that incorporates topic distributions to learn latent factors from reviews of users or items. VBPR[2] is the Visual Bayesian Personalized Ranking method that builds upon the BPR model but incorporates visual product knowledge. CKE or Collaborative Knowledge base Embedding[8] is a modern neural recommender system based on a joint model integrating matrix factorization and heterogeneous data formats, including textual contents, visual information and a structural knowledge base to infer the top-N recommendations results.

Table 1 presents the results on a number of metrics, including NDCG, Recall, HitRate (HR) and Precision. We observe that the proposed method gives statistically significant improvements over all baselines considered. Our findings suggest that including sequential history of user actions as well as hierarchical softmax function in the RL framework is indeed benefecial and helps in increasing precision and ranking metrics.

REFERENCES

[1] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences. In *The World Wide Web Conference*. ACM, 151–161.

[2] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[3] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 165–172.

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[5] Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*. Vol. 2. MIT press Cambridge.

[6] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 1835–1844.

[7] Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. *arXiv preprint arXiv:1906.05237* (2019).

[8] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 353–362.

[9] Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192* (2018).