SCIENCE CHINA
Information Sciences

• LETTER •

June 2020, Vol. 63 169101:1–169101:3
https://doi.org/10.1007/s11432-018-9820-3

Reinforcement learning with actor-critic for knowledge graph reasoning

Linli ZHANG[1,2], Dewei LI[1,2*], Yugeng XI[1,2] & Shuai JIA[1,2]

[1]*Department of Automation, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[2]*Key Laboratory of System Control and Information Processing of Ministry of Education, Shanghai* 200240*, China*

Received 6 June 2018/Revised 9 August 2018/Accepted 31 January 2019/Published online 9 May 2020

Citation Zhang L L, Li D W, Xi Y G, et al. Reinforcement learning with actor-critic for knowledge graph reasoning. Sci China Inf Sci, 2020, 63(6): 169101, https://doi.org/10.1007/s11432-018-9820-3

Dear editor,

In recent years, with the development of artificial intelligence and deep learning techniques, knowledge graph (KG) is receiving unprecedented attention. Essentially, KG is a semantic network, which stores massive information structured as entity-relation pairs in a graphical model. When the keywords or questions that users input are mapped into KG somehow, relevant information may be presented through fact triples. Therefore, large scaled KGs have wide practical application scenarios in natural language processing (NLP) tasks such as Q&A, retrieval, learning, and inference. However, the information in KG is often incomplete due to the lack of various important relations, which significantly influences KG applications in NLP tasks. Consequently, it is necessary and in great demand to do KG completion [1, 2]. Hence, how to make automated path reasoning and obtain a new relation of an arbitrary entity pair from existed evidence have aroused considerable interests among the researchers in this area.

The path-ranking algorithm (PRA) [3] is a primary path-finding approach, which learns a weighted combination of path-constrained random walks. Based on it, some modifications have been made to the PRA like synthesizing recursive random walks [4]. But training too many separate models and parameters causes it does not scale. Bordes et al. [5] proposed translation-based knowledge based method to represent every element in KG into a relatively low dimensional embedding vector space. All low dimensional entity vectors and relation matrices define the continuous state space of KG. Thus, similar entities and relations are easy to be evaluated and compared, given the previous work PRA, which operates in a fully discrete space. More recently, Xiong et al. [6] proposed a new method called DeepPath, which is the first attempt to consider RL methods for learning relational paths in KG guided by policy gradient. The supervised policy is trained with a predefined reward function so as to control multi-hop reasoning. Though it is superior to previous methods on some datasets empirically, only using policy evaluation may be high variance and inefficient.

Our contribution. In this study, we consider the problem of finding the possible relation paths of entity pairs and reasoning in a large scale KG. In contrast to DeepPath, we utilize an RL framework with actor-critic network, constructed by a policy-based network and a value-based network to learn policy and value function respectively. Since reasoning in KG requires access to many fact triples, i.e., nodes and edges in sequence, such a process of finding answers can be regarded as a serialized decision problem.

To this end, we model the external interactive environment as a Markov decision process (MDP). Obtained from the MDP environment, the current state and target state are employed to select the desired action according to the probability distri-

* Corresponding author (email: dwli@sjtu.edu.cn)

bution, which is the output of the policy network. In addition, the performance is evaluated by value network concerned with state and action. Based on RL framework, our learning method uses the idea of actor-critic [7] for reference that combines policy gradients and value estimation to train path finding process. The MDP environment and the actor-critic network make up our whole RL model. In order to predict the relation of a given entity pair, our model attempts to find reliable reasoning paths connecting it from all existed relations. Experimentally, we show that our proposed method achieves state-of-the-art performances on NELL-995 dataset [8].

MDP environment. The first part of our model is the external environment. We consider the path reasoning in KG as an MDP, which represents as a quadruple $(S, A, P(\cdot,\cdot), R(\cdot,\cdot))$. S represents the set of all entities in KG which is a continuous state space. Since each entity is a discrete node, we use translation-based embeddings [5] to model all entities to a low dimensional vector space as pretreatment. This embeddings map can reduce computation cost and improve efficiency while capturing and retaining the semantic information. A denotes the set of all relations in KG which is the action space. Given two entities, our learning method selects available actions to search relation paths linking the entity pair. $P_a(s, s')$ is the transition probability matrix and $R_a(s, s')$ is the reward function representing the timely reward of transition. In our model, we assume that we will always gain the same state s' in state s if we choose action a. The key problem of MDP is to find the desired policy: there exists a function π that chooses an action in every state, which will maximize accumulated reward. The policy output results are partly randomized and partly controlled.

Actor-critic network. The second part, actor-critic network, combines value-based and policy-based networks. Actor is a policy network that can learn stochastic policies, which is more effective in high-dimensional or continuous action space and better convergence properties. Critic is a value network that evaluates these policies, which can reduce the variance of policy evaluation and avoid converging to a local rather than global policy optimum. Actor-critic network combines two sides for complementarity in their advantage and enhances the accuracy of fact prediction. In terms of critic's value, actor updates policy to gain higher scores. Similarly, critic adjusts value by true rewards of policy chosen by actor. We use two fully-connected neural networks to represent actor and critic to parameterize the policy and value function, respectively. The actor network, policy function $\pi(s;\theta)$, inputs state vector s and outputs a probability distribution over all possible actions and obtains timely rewards. Meanwhile, the critic network, advantage function $A^{\pi}(s, a)$, maps state vector s and action a together to A-value to assess actor's performance. A-value is positive if it is better than average and vice versa. The advantage function is given by

$$A^{\pi}(s,a) = Q(s,a) - V(s) = r + \gamma V(s') - V(s), \quad (1)$$

where $V(s) = E_{\pi(s)}[r + \gamma V(s')]$ is the value function and $Q(s, a) = r + \gamma V(s')$ is the action value function. At the beginning, policy and value are both randomly chosen. During the interaction of designed MDP environment and actor-critic network, it accumulates experience and optimizes network with updated parameters. Compared to DeepPath, our network learns not only policy but also value function.

Training. Firstly, we use all positive samples for each relation to train actor-critic network with breadth-first search (BFS). We update parameters θ to maximize the expected cumulative reward using $J(\pi) = E_{a\sim\pi(s)}[V(s)]$. For operating each path found by BFS to the objective function $J(\pi)$, the approximate gradient used to update the network is shown below:

$$\nabla_{\theta} J(\pi) = E_{s\sim\rho^{\pi}, a\sim\pi(s)}[\nabla_{\theta} \log\pi(a|s) \cdot A^{\pi}(s,a)]. \quad (2)$$

This implies that likelihood of actions better than average is increased, and likelihood of actions worse than average is decreased. Secondly, we retrain the network partly controlled by the pre-trained policy and reward function. For each entity pair, we choose three possible relations to extend its reasoning path and keep the best one, which consists of top two probability relations and a randomized one relied on the action probability distribution $\pi(a|s)$. We repeat the above mentioned steps until reach the target entity or limited steps. If it successes, the parameters are updated using the following gradient:

$$\nabla_{\theta} J(\pi) = \nabla_{\theta} \sum_t \log\pi(a = r_t|s_t) \cdot R_{\text{total}}, \quad (3)$$

where $R_{\text{total}} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$ is the sum of timely reward multiplied by the decay factor and r_t belongs to the path p. Otherwise, the episode ends if it fails to reach the target entity within max_length steps. The detail of the train process is shown in Algorithm 1.

Table 1 Fact prediction results (MAP)[a)]

Tasks	ACRL	DeepPath	TransE	TransD	Improvement (%)
personBornInLocation	**0.4878**	0.2895	0.2652	0.0924	+68.50
athletePlaysForTeam	**0.4384**	0.2435	0.1400	0.1494	+80.04
teamPlaysSport	**0.4120**	0.3084	0.3567	0.1216	+33.59
agentBelongsToOrganization	0.3287	**0.3308**	0.3498	0.1286	−0.64
athletePlaysInLeague	**0.5199**	0.5059	0.4676	0.0762	+2.77
organizationHeadQuarteredInCity	**0.6420**	0.5929	0.2569	0.1520	+8.28
organizationHiredPerson	**0.5257**	0.4475	0.3073	0.3554	+17.47
...					
Overall	**0.5170**	0.4638	0.3622	0.1720	+11.47

a) The bold number represents the best result among the four methods for every task.

Algorithm 1 Training procedure

Initialize parameters θ of actor-critic network;
for episode $\Leftarrow$ 1 to N **do**
  Initialize entity pair $\langle e_1, e_2\rangle$;
  while num_path < max_path **do**
    Randomly BFS, obtain $\langle e, r\rangle$;
    if $r \neq \emptyset$ **then**
      Embed $\langle e, r\rangle$ to $\langle s, a\rangle$;
      Save $\langle s, a\rangle$ to ε_{pos};
    end if
  end while
  for $\langle s, a\rangle$ in ε_{pos} **do**
    $k \propto \nabla_\theta \log\pi(a|s) \cdot A^\pi(s, a)$;
  end for
end for
for episode $\Leftarrow$ 1 to N **do**
  Initialize state vector s_0;
  while num_step < max_step **do**
    $a_1, a_2 \sim \pi(a|s_0)_{\text{top2}}$, $a_3 \sim \pi(a|s_0)$;
    Choose the best action $a \sim R_t$;
    Save $\langle s, a\rangle$ to $\kappa_{\text{pos}}/\kappa_{\text{neg}}$;
  end while
  for $\langle s, a\rangle$ in κ_{pos} **do**
    $k \propto \nabla_\theta \sum_t \log\pi(a = r_t|s_t) \cdot R_{\text{total}}$;
  end for
  for $\langle s, a\rangle$ in κ_{neg} **do**
    $k \propto \nabla_\theta \sum_t \log\pi(a = r_t|s_t) \cdot (-1)$;
  end for
end for

Experiment. In order to evaluate our reasoning model, we conduct the experiment of the fact prediction task on NELL-995 dataset [6]. Given some positive and negative samples in each relation, the procedure predicts if an unknown fact holds. We compare our method (ACRL) with DeepPath and other methods. Table 1 shows the mean average precision (MAP) and the improvement achieved by ACRL compared with DeepPath partly. Since ACRL can find a more compact and correct set of learned paths in most tasks, actor-critic network is more efficient than policy network to guide the learning network to capture the accurate relation paths. The whole tasks MAP results and detailed experimental result analysis are given in Appendixes A and B.

Conclusion. The actor-critic network with RL framework method we proposed aims to train generative reasoning nets for relation learning and pathing effectively. Experimentally, we explicitly illustrate the superiority of our method. For future work, we plan to build clusters for correlative entities and relations to reduce the dependence on large scale training datasets.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61590924, 61433002) and Science and Technology Innovation Action Plan Project of Shanghai Science and Technology Commission (Grant No. 18511104200).

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Wang Q, Liu J, Luo Y F, et al. Knowledge base completion via coupled path ranking. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016. 1308–1318
2 Chen W H, Xiong W H, Yan X F, et al. Variational knowledge graph reasoning. 2018. ArXiv:1803.06581
3 Ni L, Cohen W W. Fast query execution for retrieval models based on path-constrained random walks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010. 881–888
4 Wang W Y, Cohen W W. Joint information extraction and reasoning: a scalable statistical relational learning approach. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, 2015. 355–364
5 Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data. In: Proceedings of Neural Information Processing Systems, 2013. 2787–2795
6 Xiong W H, Hoang T, Wang W Y. DeepPath: a reinforcement learning method for knowledge graph reasoning. 2017. ArXiv:1707.06690
7 Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms. In: Proceedings of the 31st International Conference on International Conference on Machine Learning, 2014. 387–395
8 Andrew C, Justin B, Bryan K, et al. Toward an architecture for neverending language learning. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, 2010