

Reinforcement Learning based Meta-path Discovery in Large-scale Heterogeneous Information Networks

Guojia Wan,^{1,2} Bo Du,^{1,2*} Shirui Pan,³ Gholamreza Haffari³

¹National Engineering Research Center for MultimediaSoftware, Wuhan University, Wuhan 430072, P. R. China.

²School of Computer Science, Wuhan University, Wuhan 430072, P. R. China.

³ Faculty of Information Technology, Monash University, Melbourne, Australia.

guojiawan@whu.edu.cn, remoteking@whu.edu.cn, Shirui.Pan@monash.edu, gholamreza.haffari@monash.edu

Abstract

Meta-paths are important tools for a wide variety of data mining and network analysis tasks in Heterogeneous Information Networks (HINs), due to their flexibility and interpretability to capture the complex semantic relation among objects. To date, most HIN analysis still relies on hand-crafting meta-paths, which requires rich domain knowledge that is extremely difficult to obtain in complex, large-scale, and schema-rich HINs. In this work, we present a novel framework, Meta-path Discovery with Reinforcement Learning (MPDRL), to identify informative meta-paths from complex and large-scale HINs. To capture different semantic information between objects, we propose a novel multi-hop reasoning strategy in a reinforcement learning framework which aims to infer the next promising relation that links a source entity to a target entity. To improve the efficiency, moreover, we develop a type context representation embedded approach to scale the RL framework to handle million-scale HINs. As multi-hop reasoning generates rich meta-paths with various length, we further perform a meta-path induction step to summarize the important meta-paths using Lowest Common Ancestor principle. Experimental results on two large-scale HINs, Yago and NELL, validate our approach and demonstrate that our algorithm not only achieves superior performance in the link prediction task, but also identifies useful meta-paths that would have been ignored by human experts.

Introduction

The complex interaction in real-world structured data, such as social networks, biological networks, and knowledge graphs, can be modeled as *Heterogeneous Information Networks* (HINs) (Sun and Han 2013), where objects and edges

*Corresponding author. This work was supported in part by the National Natural Science Foundation of China under Grants 61822113, the National Key R&D Program of China under Grant 2018YFA060550, the Natural Science Foundation of Hubei Province under Grants 2018CFA050 and the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170. G. H. is partly supported by a Future Fellowship from the Australian Research Council (FT190100039).

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

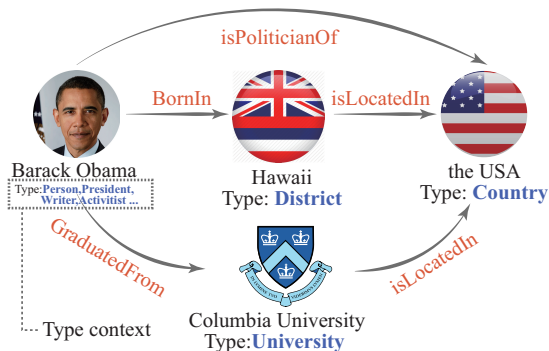


Figure 1: A HIN with multiple types and multiple relations.

are annotated with multiple types. Due to their capability to retain the rich and complex inter-dependency between objects, HIN have recently attracted increasing research attention. However, the heterogeneity and complexity of a HIN also impose significant challenges when engaging in relation analysis among objects, particularly in large-scale networks. In order to cope with these restrictions, the concept of *meta-path* were proposed to capture the semantic relation between objects (Sun et al. 2009; 2011).

Figure 1 illustrates examples meta-path between two objects, *Barack Obama* and *the USA*. To understand the relation between a person and a country, we can exploit the following meta-paths: a) Person $\xrightarrow{\text{isPoliticianOf}}$ Country, b) Person $\xrightarrow{\text{BornIn}}$ District $\xrightarrow{\text{isLocatedIn}}$ Country, c) Person $\xrightarrow{\text{GraduatedFrom}}$ University $\xrightarrow{\text{isLocatedIn}}$ Country. Although they are different in length and involve different intermediate objects, these meta-paths all help inference the semantic relation of *isCitizenOf(Person, Country)*. Thus, it is easy to deduce from Figure 1 that *Barack Obama* is a citizen of *the USA*. As meta-paths carry rich semantic information, they have been widely used in many data mining and network analysis tasks (Shi et al. 2016).

Most existing meta-path studies or meta-path guided research stipulate to predefine enumerable sets of meta-paths, which largely depends on domain experts and is labour-

intensive, as finding interesting meta-paths in HINs is very challenging. In general, this can be considered as a search problem (Lao and Cohen 2010). More specifically, given a type set T , a relation set \mathcal{R} , and a fixed length l , the possible meta-paths are in the search space of size $|T| \times (|T| \times |\mathcal{R}|)^l$. Such a huge space can result in combinatorial explosion as the scale of $|T|$, $|\mathcal{R}|$ and l increase. Most existing approaches based on human-defined meta-paths are feasible only on schema-simple HINs, e.g. DBLP (Ley 2002). Once the schema of a HIN is large and complex, it is infeasible to predefine sufficient meta-paths, particularly long ones, leading to degenerated performance in many HIN analysis tasks. Therefore, it is imperative to develop an appropriate strategy to discover meaningful meta-paths, which remains a challenge in this area.

Some previous research has aimed to automatically discover meta-paths. Graph traversal methods, such as breadth-first search (BFS) (Kong et al. 2012), A* algorithm (Zhu et al. 2018), have been used to find the shortest path in a schema-simple HIN. But they are hard to deal with a complex and large-scale HIN. Meng et al. (2015) proposed a greedy algorithm named *FSPG* to discover the most relevant meta-paths iteratively. However, *FSPG* operates in a fully discrete space, which makes it difficult to evaluate and compare similar objects and relations in a HIN.

Recently, multi-hop reasoning (Shen et al. 2018; Xiong, Hoang, and Wang 2017), has emerged as a promising approach in inferring paths linking two objects in a knowledge graph that also considered as a type of semantic rich HIN. This approach involves sampling the most promising relation (edge) to extend a path from a source node to a target node. However, this approach has the following limitations: 1) it is not an end-to-end reasoning approach and it heavily relies on the precomputed entity embedding, which is typically learned by a translation-based embedding method (e.g., TransE (Bordes et al. 2013)), so that the reasoning state can be represented in a continuous space and be used in a reinforcement learning (RL) agent. Employing an embedding learning approach before reasoning is not only time-consuming but also increases memory consumption, hindering its ability to handle million-scale HINs. 2) These algorithms generate only bare paths linking two objects, without meta-path induction (summarization) for downstream tasks such as link prediction. 3) These approaches ignore different type information in an entity, which however is very important, and can provide rich interpretation of the relationships among objects in HINs.

Based on the above observations, we here propose a novel reinforcement learning framework to automatically mine interesting meta-paths from large-scale HINs, named as Meta-path Discovery with Reinforcement Learning (MPDRL). Our goal is to employ a reinforcement learning agent to infer the most meaningful paths from a source object to a target object and then to perform a further meta-path induction step to summarize the meta-paths from a large number of generated paths. Our approach has three desirable properties. First, our reasoning framework requires no pre-training and other supervision or fine-tuning information over prior knowledge. Second, the proposed approach has the built-

in flexibility to consider interpretable meta-paths of varying lengths, which is important for inferring long-range meta-paths. Third, the agent can identify diverse objects by their type context, allowing the system to be run successfully on million-scale HINs. Our approach is applied on two HINs with complex schema, Yago and NELL, and has yielded rich meta-paths by the agent’s multi-hop reasoning. Moreover, the experimental results of link prediction demonstrate that our approach out-performs the compared approaches.

Our contribution are three-fold:

- We present an RL-based framework, MPDRL¹, to mine meta-paths from large-scale complex Heterogeneous Information Networks without the need of human effort.
- We propose type context representation embedded approach in our RL approach, and design a policy network to memorize or forget historical state, allowing our proposed algorithm to easily handle million-scale HINs.
- We conduct the link prediction task based on extracted meta-paths on two large-scale and complex HINs, Yago and NELL. Experimental results demonstrated that our algorithm not only reveals the synonymous meta-paths, but also outperforms approaches employing meta-paths designed by human experts.

Related work

Meta-path guided approaches To analyze and perform data mining tasks in HINs, Sun et al. (2009; 2011) proposed the concept of meta-paths to capture semantic information and express complex relevance of two objects. Subsequently, a number of papers have been published that involve meta-paths in many data mining tasks in HINs, such as similarity measurement (Sun et al. 2011; Wang et al. 2016), link prediction (Shi et al. 2014; Cao, Kong, and Philip 2014), representation learning (Dong, Chawla, and Swami 2017; Cao, Kong, and Philip 2014), and so on.

Discovery meta-paths in HINs Many meta-path guided approaches suffer from a major drawback, i.e., they require domain experts to manually predefine a series of meta-paths. Lao and Cohen (2010) proposed a method based on random walk to discover and leverage meta-paths of labeled relational networks within a fixed length l . However, l is hard to set, as it varies among different datasets. A recent study by Meng et al. (2015) developed a greedy algorithm named *FSPG* to discover the most relevant meta-paths, further developing a *Greedytree* data structure to find meta-paths iteratively. Yang et al. (2018) pointed out that the path-finding process is a combination problem. Thus, they proposed a similarity measurement model that can predefine meta-paths by reinforcement learning. However, this approach is only effective on schema-simple HINs. Shi and Wenginger (2014) discussed meta-path discovery at various levels of type-granularity in a large complex HIN and proposed a general framework to mine meta-paths from complex HINs using adaptations from classical knowledge discovery techniques.

Multi-hop reasoning in graphs Many multi-hop reasoning approaches (Lao, Mitchell, and Cohen 2011) based on ran-

¹<https://github.com/mxz12119/MPDRL>

dom walk have been proposed to capture more complex reasoning patterns in a knowledge base. However, the reasoning paths gathered by performing random walks are independent of the type of objects. Recently, deep reinforcement learning has achieved great success in many artificial intelligence problems (Mnih et al. 2015). Deep reinforcement learning allows a policy function to be learned from graph-based data for multi-hop reasoning. Xiong et al. (2017) investigated the multi-hop reasoning with RL on knowledge bases. However, it also ignores the types of objects. Das et al. (2018) and Shen et al. (2018) further studied reinforcement learning for knowledge base completion.

Definitions and annotations

Definition 1 (Heterogeneous Information Network) A HIN is an information network with multiple types of nodes and edges, which defined as a graph $G = (\mathcal{V}, E)$. \mathcal{V} denotes an object set with a type mapping function: $\phi : \mathcal{V} \rightarrow T$, where T denotes a type set. E denotes an edge set with a relation mapping function: $\varphi : E \rightarrow \mathcal{R}$, where \mathcal{R} is a relation set. A node denotes to an object $v \in \mathcal{V}$. An edge describes the relation $r \in \mathcal{R}$ between two objects.

Example: Figure 1 illustrates a HIN, including $T = \{Person, President, Writer, Activist, University, District, Country\}$, $\mathcal{R} = \{isPoliticianOf, BornIn, isLocatedIn, GraduatedFrom\}$.

Definition 2 (Meta-path) Given a HIN G , a meta-path Ω is defined as a sequence in the form of $\omega_1 \xrightarrow{r_1} \omega_2 \xrightarrow{r_2} \dots \xrightarrow{r_n} \omega_{n+1}$, where ω_i denotes the type of the object, $\omega_i \in T$, and $r_i \in \mathcal{R}$ denotes the relation.

Example: From the path instance Barack Obama $\xrightarrow{isPoliticianOf}$ the USA, we can derive the meta-path: Person $\xrightarrow{isPoliticianOf}$ Country. A meta-path can measure the closeness between objects and guide modeling the similarity computation.

Methodology

Overview of MPDRL A schematic overview of our proposed approach is presented in Figure 2. MPDRL aims to discover meta-paths from a HIN via multi-hop reasoning between objects; this process consists of two steps.

- **Multi-hop Reasoning with RL for Path Instance Generation:** The RL agent carries out multi-hop reasoning and generates various path instances. The agent starts at the source objects, *Obama* and *Trump*. It then observes the current state and decides to move to the next object that has the highest probability of reaching the target object via the policy network. The agent alternates between such observation and movement until reaching the target object or the maximum length, thus generating a trajectory. The trajectory of an episode is a path instance for the topic *isCitizenOf*. The reasoning process can be formalized as a Markov Decision Process (MDP), and the agent can be trained by reinforcement algorithms.
- **Meta-path Induction from Path Instances:** We further refine and summarize these path instances by searching

Table 1: Annotation table

Symbol	Meaning	Symbol	Meaning
\mathcal{V}	Object set	v	Object
\mathcal{R}	Relation set	r	Relation
T	Type set	ω	Type
v_0	Source object	v_d	Target object
G	HIN	E	Edge set
S	State set	s	State
\mathcal{A}	Action set	a	Action
R	Reward	γ	Reward factor
π	Policy function	θ	Parameters
τ	Agent trajectory	\mathbf{h}_t	History vector
\mathbf{z}	Update gate	\mathbf{q}	Reset gate
\circ	Hadamard product	f	Full-connected layer
σ	Sigmoid function	\mathcal{G}	DAG
l	Maximum length	Ω	Meta-path

We denote a vector using a bold letter, e.g. \mathbf{v} corresponding to v

for the Lowest Common Ancestor (LCA) in type directed acyclic graph (DAG) so as to generate various meta-paths.

Multi-hop Reasoning with RL for Path Instance Generation

In the below, we describe in detail our multi-hop reasoning approach with reinforcement learning.

Reinforcement learning architecture in HINs Reinforcement learning follows a Markov Decision Process (MDP) formulation, which an agent learns from the interactions with the environment derived from a HIN through sequential exploration and exploitation. In a HIN, we formalize RL with the quartuple $(S, \mathcal{A}, \mathcal{P}, R)$, whose elements are elaborated below.

States. The state s_i at step i is defined as a tuple (v_{i-1}, r_i, v_i, v_d) , where $v_i \in \mathcal{V}$ is the current object, v_{i-1} is the last object, r_i denotes the relation between v_i and v_{i-1} , and v_d is the target object. $s_i \in S$, where the state space S consists of all valid combination in $\mathcal{V} \times \mathcal{R} \times \mathcal{V} \times \mathcal{V}$. Given a path-finding pair (v_0, v_d) , the starting state is represented as $(\text{'ST'}, \text{'ST'}, v_0, v_d)$, where a start state indicator 'ST' was added to indicate the initial state of the agent. The terminal state is (v_{t-1}, r_t, v_d, v_d) . Each state captures the agent's position in the HIN. After taking an action, the agent will move to the next state.

Actions. The action space A_{s_i} for a state $s_i = (v_i, r_{i+1}, v_{i+1}, v_d)$ is the set of outgoing edges of the current object v_i in the HIN, where $A_{s_i} = \{(r, v) | (v_i, r, v) \in G, v \notin \{v_0, v_1, \dots, v_d\}\}$. Beginning with the source object v_0 , the agent uses the policy network to predict the most promising path, then extends its path at each step until it reaches the target object v_d .

Transition. The transition \mathcal{P} is the state transition probability used to identify the probability distribution of the next state, which is defined as a map function: $\mathcal{P} : S \times \mathcal{A} \rightarrow S$. The policy network encodes the current state to output a probability distribution $\mathcal{P}(s_{i+1} | s_i, a_i)$, where $a_i \in A_{s_i}$. In our RL framework, the transition strategy involves selecting the action with maximum probability in A_{s_i} .

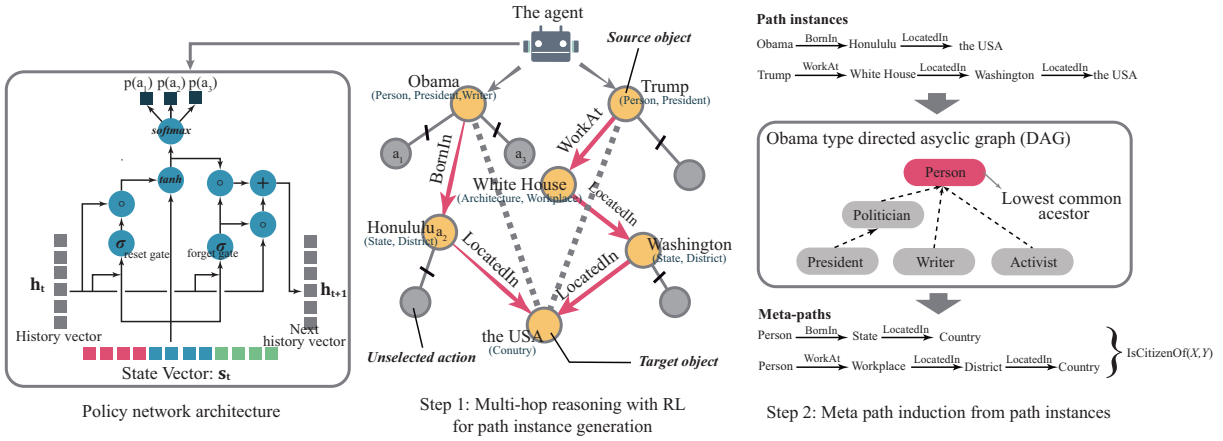


Figure 2: Overview of MPDRL. MPDRL consists of two step: 1) Multi-hop reasoning with RL for path instance generation in the HIN, 2) Meta-path induction from path instances. The left gray box is the architecture of our policy network.

Rewards. Given a pair (v_0, v_d) , if the agent reaches the target object, i.e. $v_i = v_d$, the agent’s trajectory is labeled as a successful finding. The reward for each hop is defined as follows:

$$R(\tau_i) = \begin{cases} 1 \cdot \gamma^i, & v_i = v_d \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where $\gamma > 0$ is the reward factor and τ_i is the i -th step of the trajectory. The reward factor is made flexible to control the trade-off between the long-term and short-term reward feedback. If $\gamma < 1$, the agent is likely to pick a short finding-path. If $\gamma > 1$, the agent will prefer a longer path.

It should be noted here that the positive rewards usually suffer from the sparsity problem, being received only reaching at the end of the correct object. To tackle this reward sparsity issue, we augment another action option, marked as ‘OP’, i.e. $\mathcal{A}_{s_i} \leftarrow \mathcal{A}_{s_i} \cup \{‘OP’\}$. ‘OP’ means that the agent fails to reach the correct object, as a result it stops and receives a negative reward. This is especially helpful for preventing the agent from getting stuck in intermediate states, thereby accelerating convergence of the training.

Policy network Due to the large search space in the complex HIN, we design a model-free policy network $\pi(s, \mathcal{A}) = P(a|s; \theta)$ based on deep learning to model the RL agent in a continuous space, where θ is the neural network parameter. Considering that the agent need to do sequential decision making, we introduce a history vector \mathbf{h}_t to keep history information in order to better guide the agent. Given a trajectory τ at step t , the history vector is determined by the last history \mathbf{h}_{t-1} and the last state \mathbf{s}_{t-1} , where $\mathbf{s}_{t-1} = [\mathbf{v}_{t-1}; \mathbf{r}_{t-1}; \mathbf{v}_d]$, while $\mathbf{v}, \mathbf{r} \in \mathbb{R}^d$,

$$\mathbf{h}_t = H(\mathbf{h}_{t-1}, \mathbf{s}_{t-1}). \quad (2)$$

Note that Eq. (2) is a recursive formula. To encode the history vector, we introduce a gating mechanism which is similar to GRU (Cho et al. 2014) shown in Figure 2 to control the memorization or forgetting of the history information .

H is defined as follows,

$$\begin{aligned} \mathbf{z}_t &= f_z(\mathbf{s}_{t-1}, \mathbf{h}_{t-1}) \\ \mathbf{q}_t &= f_q(\mathbf{s}_{t-1}, \mathbf{h}_{t-1}) \\ \tilde{\mathbf{h}}_t &= f_h(\mathbf{s}_{t-1}, \mathbf{h}_{t-1} \circ \mathbf{z}_t) \\ \mathbf{h}_t &= \mathbf{q}_t \circ \tilde{\mathbf{h}}_t + (1 - \mathbf{q}_t) \circ \mathbf{h}_{t-1}, \end{aligned} \quad (3)$$

where $\mathbf{z}_t \in \mathbb{R}^d$ is the update gate, $\mathbf{q}_t \in \mathbb{R}^d$ is the reset gate, \circ is the Hadamard product, $\tilde{\mathbf{h}}_t$ denotes the hidden layer, $[\cdot; \cdot]$ denotes the concatenation operation, and f is the full-connected layer with an activation function. Based on the GRU-like recurrent cell architecture, the history vector is updated according to the agents movement dynamics. Moreover, unlike the classic GRU cell, which uses \mathbf{h}_t to predict y , we find that $\tilde{\mathbf{h}}_t$ works better in the HIN environment. Therefore, the distribution of y , i.e. all possible actions, is defined as follows,

$$\mathbf{a} = \text{softmax}(\text{RELU}(\mathbf{W}_U \tilde{\mathbf{h}}_t + \mathbf{W}_V \mathbf{s}_{t-1} + \mathbf{b})), \quad (4)$$

where $\mathbf{a} \in \mathbb{R}^{|\mathcal{A}|}$ denotes the probability distribution for all actions. As a result, the agent picks the action with the maximum probability then moves to the next state.

Type context learning for object representation Due to the large scale and complicated semantic context of the objects in HINs, it is challenging to model each object in the states. One solution is to use pre-training embeddings or content information to represent objects and relations. However, before obtaining the pre-training, most embedding learning approaches in HINs still require well-defined meta-paths (Fu, Lee, and Lei 2017). The other issue would be to initialize the objects and relations with a finite dimensional vector. However, it vastly increase the required memory storage when the number of objects is million-scale.

Generally, an object is associated with a type set, e.g. $Obama: \{\text{Writer, President, Activist, Person}\}$, $Trump: \{\text{President, Businessman, Person}\}$, while $Obama$ is the type-sibling of $Trump$ because they share the same

type *President*. The type information not only explicitly expresses the context of an object in a HIN, but also reveals the inner relevance among objects. Therefore, learning type context allows the agent to efficiently identify the context of an object in a HIN. Some recent works have placed the type information into knowledge graphs or information networks and thereby achieved improved results in experiments (Xie, Liu, and Sun 2016). Based on this observation, we propose a simple but effective approach to modeling objects in states, which contributes to enables our RL approach to deal with large-scale HINs. For an object v with a type set $T_v \subset T$, the type context representation of v is defined as follows,

$$\mathbf{v} = \frac{1}{|T_v|} \sum_{\omega_i \in T_v} \omega_i, \quad (5)$$

where $\omega_i \in \mathbb{R}^d$ is the i -th type vector in T_v . To summarize, the position of an object is determined by its type context.

Meta-path Induction from Path Instances

In this section, we discuss how to generate meta-paths from a path instance. The proposed RL approach can train an agent capable of multi-hop reasoning to automatically pick path instances between object pairs (v_0, v_d) . A path instance, i.e. a trajectory, is in the form $v_1 \xrightarrow{r_1} v_2 \xrightarrow{r_2} \dots \xrightarrow{r_n} v_{n+1}$. The previous works assume that the objects in a HIN only are only of one type (Shi and Weninger 2014). As a result, the meta-paths are generated via simple replacement. However, in a large-scale HIN, an object normally has multiple types. Therefore, simple replacement will bring about a number of low-relevance meta-paths.

Consequently, the reduction of the type set is necessary for assigning a type to an object. Generally, the type structure in a HIN is organized in the form of a directed acyclic graph (DAG) $\mathcal{G}_T = (T, E)$, where T is a set including whole types, while E denotes the directed link between two types. The edges of the DAG are assumed to point from parents to children, e.g. the type *president* is the subordinate type of *people*. Step 2 in Figure 2 shows a toy example of a type DAG. To assign a type to an object that is associated with a type set T_v , we choose the Lowest Common Ancestors (LCA) of \mathcal{G}_{T_v} . Specifically, we employ a naive LCA algorithm in (Bender et al. 2005) to find nearest types that close to root type so as to obtain the key types.

In the naive LCA algorithm, it first traverses the DAG in a breadth-first manner and assigns depth to every node, then simply walks up the DAG to find the ancestors of the queried nodes, from which it chooses the node of greatest depth. The input of the naive LCA is a DAG \mathcal{G}_{T_v} , while the output is a set containing several key types. Finally, we can identify meta-paths through identifying all valid combinations among the output types and relations.

Optimization and training

The objective function of the policy network is to maximize the expectation of long-term accumulated rewards,

$$J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} [R(\tau)], \quad (6)$$

where τ denotes an N -length trajectory generated from the underlying distribution $p_\theta(\tau)$, while $R(\tau)$ is the reward function for τ . To optimize the above objective function, we use the policy gradient to maximize $J(\theta)$. The gradient of $J(\theta)$ is as follows,

$$\begin{aligned} \frac{\partial J(\theta)}{\partial \theta} &= \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\left(\frac{\partial}{\partial \theta} \log p_\theta(\tau) \right) R(\tau) \right] \\ &= \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\left(\sum_{i=1}^N \frac{\partial}{\partial \theta} \log \pi_\theta(a_i | s_{i-1}, a_{i-1}) \right) R(\tau) \right], \end{aligned} \quad (7)$$

where $p_\theta(\tau)$ can be decomposed to the process of $\pi_\theta(a_i | s_{i-1}, a_{i-1})$. In order to estimate the π_θ , moreover, we can compute an approximation of $J(\theta)$ by averaging the accumulated rewards on a series of trajectories generated from the interaction between the agent and the environment by *REINFORCE* (Williams 1992),

$$\frac{\partial J(\theta)}{\partial \theta} \approx \frac{1}{K} \sum_{j=1}^K \left[\sum_{\tau^j, i=1}^{N^j} \frac{\partial}{\partial \theta} \log \pi_\theta(a_i^j | s_{i-1}^j, a_{i-1}^j) \gamma^i \right]. \quad (8)$$

To improve the training efficiency, we limit the maximum length of a trajectory to l . When the agent reaches l , the finding process is stopped and a negative reward is returned. We employ *ADAM Optimizer* (Kingma and Ba 2014) to optimize the policy network. The parameter θ is updated every k episodes.

Experiments

Experimental settings

For the purpose of validating the efficiency and effectiveness of our approach MPDRL, we perform link prediction based on the generated meta-paths. Link prediction provides us with a measurable and objective way of evaluating our discovered meta-paths (Meng et al. 2015).

Datasets We perform experiments on two online HINs as well as knowledge bases, Yago and (Suchanek, Kasneci, and Weikum 2007) and NELL (Mitchell et al. 2018), which contain more complex type mapping and relation mapping comparing to schema-simple HINs.

- **Yago** is a large-scale knowledge base derived from Wikidata, WordNet, and GeoNames (Suchanek, Kasneci, and Weikum 2007). We here use the ‘CORE facts’ portion of Yago², which consists of 12 million facts, 4 million entities, 0.8 million types, and 38 relations.
- **NELL** is a knowledge base that extracted from the Web text of more than one billion documents. We here use the 1,115-th dump of NELL³, which consists of 2.7 million facts, 2 million entities, 758 types, and 833 relations. We then remove the triples with the relation *generalizations* as this relation describes redundant object type information that is already included.

Baselines: *HeteSim* (Shi et al. 2014; 2012), *FSPG* (Meng et al. 2015), *PCRW* (Lao and Cohen 2010), *AutoPath* (Yang et

²<https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/>

³<http://rtw.ml.cmu.edu/rtw/>

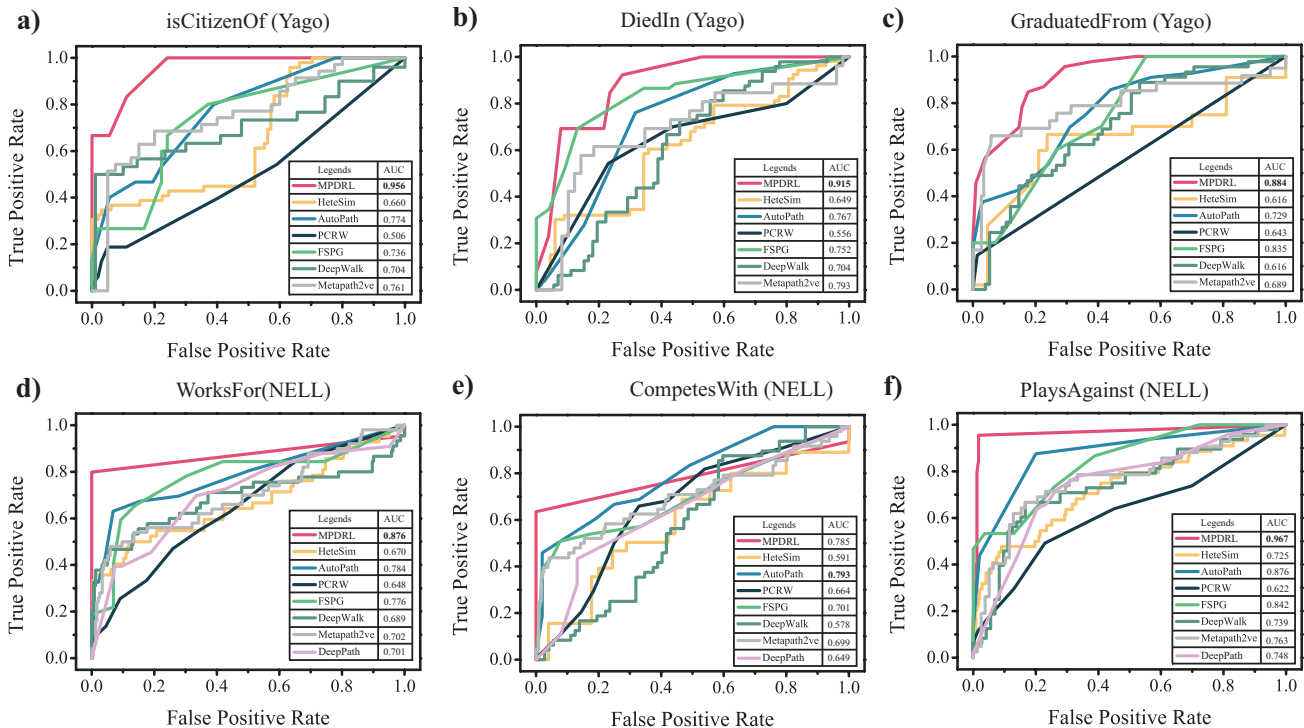


Figure 3: ROC and AUC for link prediction on six topics: a) isCitizenOf(Yago), b) DiedIn(Yago), c) GraduatedFrom(Yago), d) WorksFor(NELL), e) CompetesWith(NELL), f) PlaysAgainst(NELL).

al. 2018), *DeepWalk* (Perozzi, Al-Rfou, and Skiena 2014), *Metapath2vec* (Dong, Chawla, and Swami 2017), *DeepPath* (Xiong, Hoang, and Wang 2017).

Note that the embedding-based approaches, i.e. *DeepWalk* and *Metapath2vec*, output object vectors. For performing link prediction on a pair, we use Hadamard product of the two object embeddings as the input of the SVM classifier.

In the RL agent training stage, the key hyper-parameter settings are as follows: maximum length l is fixed to 5, learning rate α is 0.005, reward factor γ is 1.5, and update frequency k is 50. Vector dimension d is 100.

Link prediction results

Three relations for each dataset, $\{isCitizenOf, DiedIn, GraduatedFrom\}$ in Yago and $\{WorksFor, CompetesWith, PlaysAgainst\}$ in NELL were evaluated for the link prediction task. For a certain relation task, e.g. *isCitizenOf*, we expect to obtain various meta-paths from the RL agent. The facts with such relation were removed from the HIN. A sample set including positive pairs and negative pairs was then constructed based on the removed facts. Positive pairs were directly derived from the removed facts. Each negative pair was generated by replacing the true target object v_d with a fake one v'_d in each pair (v_α, v_d) , where v'_d has the same types as v_d . Finally, we adopt a linear regression model (Lao and Cohen 2010) with $L1$ regularization to perform link prediction using the binary meta-path features in order to predict whether the relation $isCitizenOf(v_\alpha, v_\beta)$ exists for a test pair (v_α, v_β) . As for the binary meta-path feature, if a

meta-path Ω connects the pair (v_α, v_β) , the value of binary meta-path feature is 1 otherwise this value is 0.

The Receiver Operating Characteristic (ROC) curves and the Area under the curve (AUC) are presented in Figure 3. As the plots show, the classifiers trained by meta-paths generated using our approach exhibit superior performance in all six relations. Although the existing embedding-based methods, such as *DeepWalk* and *Metapath2Vec*, made significant progress on representation learning on HINs, we found that once the HIN becomes complex, the heterogeneity causes performance drop, as evidenced by *DeepWalk*. Moreover, the performance of *Metapath2Vec* is also hindered by the limited hand-crafting meta-paths. Additionally, when two objects are connected by a longer path, as in the case of *GraduatedFrom*, such models are worse. *FSPG*, as well as *AutoPath* considers various meta-paths generated by themselves and thus perform better than *PCRW* and *PCRW*, indicating that more effective meta-paths results in a better performance. Furthermore, the superior performance achieved by our RL agents demonstrates its ability to generate many effective meta-paths.

Meta-paths Analysis

For each relation, the trained Linear regression with $L1$ regularization, also known as Lasso regression, can perform subset selection and return the coefficients of meta-paths. These coefficients imply the weight of meta-paths. As a result, we present the five most relevant meta-paths generated by our RL agent in Table 2 and 3. As the results show, our RL

Table 2: Example meta-paths found by our RL model from four topics in Yago and NELL.

Relation	Meta-path	Weight
isCitizenOf (Yago)	Person $\xrightarrow{\text{BorIn}}$ Country	0.561
	Person $\xrightarrow{\text{BorIn}}$ District $\xrightarrow{\text{LocatedIn}}$ Country	0.232
	Company $\xrightarrow{\text{HasOffice}}$ City $\xrightarrow{\text{HasCompany}}$ Company	0.101
	Person $\xrightarrow{\text{DiedIn}}$ Country	0.060
	Person $\xrightarrow{\text{PlaysFor}}$ Club $\xrightarrow{\text{LocatedIn}}$ Country	0.045
GraduatedFrom (Yago)	Person $\xrightarrow{\text{WorksAt}}$ University	0.734
	Person $\xrightarrow{\text{WorksAt}}$ Institution	0.112
	Person $\xrightarrow{\text{Influences}}$ Person $\xrightarrow{\text{WorksAt}}$ University	0.069
	Person $\xrightarrow{\text{hasAcademicAdvisor}}$ Person $\xrightarrow{\text{WorksAt}}$ University	0.045
CompetesWith (NELL)	Person $\xrightarrow{\text{MarriedTo}}$ Person $\xrightarrow{\text{WorksAt}}$ University	0.035
	Company $\xrightarrow{\text{HasOffice}}$ City	0.632
	Company $\xrightarrow{\text{HeadQuartredIn}}$ City	0.180
	Company $\xrightarrow{\text{HasOffice}}$ City $\xrightarrow{\text{HasCompany}}$ Company	0.095
	Company $\xrightarrow{\text{HeadQuartredIn}}$ City $\xrightarrow{\text{LocatedIn}}$ City	0.051
PlaysAgainst (NELL)	Company $\xrightarrow{\text{HasOffice}}$ City $\xrightarrow{\text{HasCompany}}$ Company $\xrightarrow{\text{LocatedIn}}$ City	0.041
	SportsTeam $\xrightarrow{\text{CompetesWith}}$ SportsTeam	0.604
	SportsTeam $\xrightarrow{\text{KnowsAs}}$ SportsTeam	0.118
	SportsTeam $\xrightarrow{\text{PlaysIn}}$ SportsTeam $\xrightarrow{\text{GameTeam}}$ SportsTeam	0.066
	SportsTeam $\xrightarrow{\text{CompetesWith}}$ SportsTeam $\xrightarrow{\text{KnowsAs}}$ SportsTeam	0.080
SportsTeam $\xrightarrow{\text{PlaysIn}}$ SportsLegue $\xrightarrow{\text{SubpartOf}}$ SportsTeam	0.013	

agent is able to find various meta-paths in a HIN. Taking the *isCitizenOf* relation as an example, the five meta-paths all obviously indicate the strong relevance between the ‘Person’ objects and ‘Country’ objects, indicating that our approach can automatically find various meta-paths without any hand-crafting process.

Interesting Meta-path discovery by our method Existing hand-crafting methods may fail to induce meta-paths like Person $\xrightarrow{\text{Influences}}$ Person $\xrightarrow{\text{WorksAt}}$ University, which is an strong indicator that a person is graduated from a university. Furthermore, our RL agent also find long-length meta-paths, such as the fifth meta-path of *CompetesWith* in NELL. Interestingly, we also found that our approach is capable of finding the hyponym relation. Taking *CompetesWith* as an example, the relation *HasOffice* that appears in the meta-path Company $\xrightarrow{\text{HasOffice}}$ City is a subordinate relation of *HeadQuartredIn*.

Effectiveness

As discussed above, the large search space is the major challenge for reasoning in a complex HIN. We deal with this problem by designing a reinforcement learning framework. To demonstrate the efficiency of our approach, we present the agent’s average reasoning success ratio and average reward within ten episodes after different numbers of training episodes. As Figure 4a shows, under $\gamma = 1.5$, we can observe that the average reasoning success ratio of *WorksFor* tends to saturate after 300 episodes, and that of *isCitizenof* does so after 500 episodes. These results indicate that our RL agent can learn the reasoning path from a given pair (v_0, v_d) . Even if the agent has not seen the objects before, it still find a promising path.

Type context learning: Moreover, to facilitate the understanding of how the type context regulates our framework,

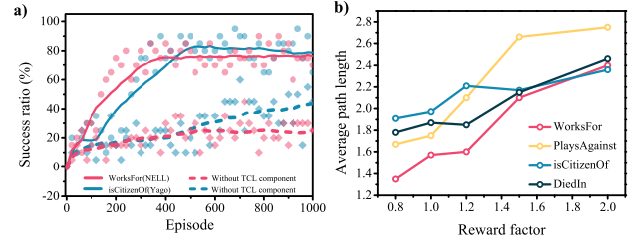


Figure 4: RL model analysis: a) Episode versus success ratio. The raw data are marked by light color dot. The curves are obtained from applying an average filter on the corresponding raw data. Dash line means the models are free of type context learning (TCL) module. b) Average path length versus reward factor.

we remove the type context representation module from our framework. Thereby, the object embeddings are initialized by a trainable embedding layer. As the green and yellow curves show, the increase of reasoning success rate without type context learning is slow, indicating a poor convergence in the training procedure. Thus, the performance gap reveals that type context learning is a key element of the superiority of our approach. The type context representation learns the type information in the HIN, meaning that the object representation can be rapidly and effective learned by sharing type representation.

Reward factor γ : To analyze how the reward factor γ influences the properties of our RL agent, we record the average reasoning path length versus reward factor during the inferring procedure. As Figure 4b shows, as the reward factor increases, the average reasoning path length rises as well, indicating γ flexibly controls the RL agent’s exploration preference. Moreover, while a larger reward factor enables the agent to find longer paths, we also found that training becomes very slow when the reward factor < 0.8 , resulting from gradient vanishing caused by the low reward feedback.

Conclusions

In this paper, we propose a RL framework that can automatically mine interesting meta-paths without any human supervision in large-scale Heterogeneous Information Networks. More specifically, we exploit the type context representation learning to scale up reinforcement learning to million-scale HINs. Unlike previous path-finding models that are operated in a discrete space, our approach allows the agent to operate multi-hop reasoning in a continuous space so as to control the distribution of the found meta-paths, thereby significantly reducing the size of large search space. These effective meta-paths can also be used in downstream HIN analysis tasks. We conducted the meta-paths mining on two HINs, Yago and NELL, yielding reasonable meta-paths on six topics. These meta-paths were further used to perform link prediction task to evaluate our model. The experimental results show that the classifiers trained by these meta-paths generally out-perform the other baselines.

References

- Bender, M. A.; Farach-Colton, M.; Pemmasani, G.; Skiena, S.; and Sumazin, P. 2005. Lowest common ancestors in trees and directed acyclic graphs. *Journal of Algorithms* 57(2):75–94.
- Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*, 2787–2795.
- Cao, B.; Kong, X.; and Philip, S. Y. 2014. Collective prediction of multiple types of links in heterogeneous information networks. In *ICDM*, 50–59. IEEE.
- Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Das, R.; Dhuliawala, S.; Zaheer, M.; Vilnis, L.; Durugkar, I.; Krishnamurthy, A.; Smola, A.; and McCallum, A. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *ICLR*.
- Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*, 135–144. ACM.
- Fu, T.-y.; Lee, W.-C.; and Lei, Z. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*, 1797–1806. ACM.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *ICLR*.
- Kong, X.; Yu, P. S.; Ding, Y.; and Wild, D. J. 2012. Meta path-based collective classification in heterogeneous information networks. In *CIKM*, 1567–1571. ACM.
- Lao, N., and Cohen, W. W. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine learning* 81(1):53–67.
- Lao, N.; Mitchell, T.; and Cohen, W. W. 2011. Random walk inference and learning in a large scale knowledge base. In *EMNLP*, 529–539.
- Ley, M. 2002. The dblp computer science bibliography: Evolution, research issues, perspectives. In *International Symposium on String Processing and Information Retrieval*, 1–10. Springer.
- Meng, C.; Cheng, R.; Maniu, S.; Senellart, P.; and Zhang, W. 2015. Discovering meta-paths in large heterogeneous information networks. In *WWW*, 754–764. International World Wide Web Conferences Steering Committee.
- Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; et al. 2018. Never-ending learning. *Communications of the ACM* 61(5):103–115.
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529.
- Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *SIGKDD*, 701–710. ACM.
- Shen, Y.; Chen, J.; Huang, P.-S.; Guo, Y.; and Gao, J. 2018. M-walk: Learning to walk over graphs using monte carlo tree search. In *NeurIPS*, 6786–6797.
- Shi, B., and Wenginger, T. 2014. Mining interesting meta-paths from complex heterogeneous information networks. In *ICDM Workshop*, 488–495. IEEE.
- Shi, C.; Kong, X.; Yu, P. S.; Xie, S.; and Wu, B. 2012. Relevance search in heterogeneous networks. In *EDBT*, 180–191. ACM.
- Shi, C.; Kong, X.; Huang, Y.; Philip, S. Y.; and Wu, B. 2014. Hetesim: A general framework for relevance measure in heterogeneous networks. *IEEE Transactions on Knowledge and Data Engineering* 26(10):2479–2492.
- Shi, C.; Li, Y.; Zhang, J.; Sun, Y.; and Philip, S. Y. 2016. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering* 29(1):17–37.
- Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *WWW*, 697–706.
- Sun, Y., and Han, J. 2013. Mining heterogeneous information networks: a structural analysis approach. *Acm Sigkdd Explorations Newsletter* 14(2):20–28.
- Sun, Y.; Han, J.; Zhao, P.; Yin, Z.; Cheng, H.; and Wu, T. 2009. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *EDBT*, 565–576. ACM.
- Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4(11):992–1003.
- Wang, C.; Sun, Y.; Song, Y.; Han, J.; Song, Y.; Wang, L.; and Zhang, M. 2016. Relsim: relation similarity search in schema-rich heterogeneous information networks. In *ICDM*, 621–629. SIAM.
- Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.
- Xie, R.; Liu, Z.; and Sun, M. 2016. Representation learning of knowledge graphs with hierarchical types. In *IJCAI*, 2965–2971.
- Xiong, W.; Hoang, T.; and Wang, W. Y. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *EMNLP*, 564–573.
- Yang, C.; Liu, M.; He, F.; Zhang, X.; Peng, J.; and Han, J. 2018. Similarity modeling on heterogeneous networks via automatic path discovery. In *ECML-PKDD*, 37–54. Springer.
- Zhu, Z.; Cheng, R.; Do, L.; Huang, Z.; and Zhang, H. 2018. Evaluating top-k meta path queries on large heterogeneous information networks. In *ICDM*, 1470–1475. IEEE.