Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 12, December 2019, pp. 3209-3218
DOI: 10.23940/ijpe.19.12.p12.32093218

KGIPSL: A Knowledge Graph Inference Method based on Probabilistic Soft Logic

Yaqiong Qiao[a], Yanjun Wang[a,b], Jiangtao Ma[a,b,*], Xiangyang Luo[a], and Huaiguang Wu[b]

[a]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, 450002, China*
[b]*Zhengzhou University of Light Industry, Zhengzhou, 450002, China*

Abstract

Knowledge graph inference has a wide range of applications in semantic search, question answering systems, entity disambiguation, link prediction, and recommendation systems. However, the accuracy and operational efficiency of existing methods do not meet the needs of large-scale knowledge graphs. Aiming at the problem of large-scale knowledge graph inference, this paper proposes a knowledge graph inference method based on probabilistic soft logic (KGIPSL). Firstly, KGIPSL uses the Markov logic network to construct the relationship between entities. Secondly, KGIPSL employs probabilistic soft logic to represent non-deterministic knowledge and infers the relationship between entities in the knowledge graph. Thirdly, KGIPSL conducts accurate knowledge inference. Experiments on real knowledge graph datasets show that the KGIPSL method is superior to the existing baseline method in accuracy, recall, and efficiency. Among them, the average accuracy of KGIPSL on the YAGO dataset is 14.9% higher than that of the baseline method.

Keywords: knowledge graph inference; probabilistic soft logic; Markov logic network

(Submitted on September 16, 2019; Revised on October 11, 2019; Accepted on November 25, 2019)

1. Introduction

The facts in knowledge graphs are characterized by inaccuracy, uncertainty, and incompleteness. Although there is a large amount of entity information in knowledge graphs, the problem of information loss is still outstanding. For example, 68% of person entities have no occupational information in Freebase knowledge graphs, 71% of person entities have missing birthplace information, 75% of person entities have no nationality information, 91% of person entities have no educational background information, and 92% of person entities have no spouse information [1]. Knowledge graph inference is based on existing knowledge in knowledge graphs to infer new and unknown knowledge, which can improve the completeness and expand the coverage of knowledge. Knowledge graph inference has a wide range of applications in semantic search, question answering systems, entity disambiguation, link prediction, and recommendation systems. However, the accuracy and operational efficiency of existing methods do not meet the needs of large-scale knowledge graphs.

This paper proposes a Markov logic network inference model KGIPSL based on probabilistic soft logic to solve the large-scale knowledge graph inference problem. Firstly, the Markov logic network is used to construct the relationship between entities. Then, the method of probabilistic soft logic is used to represent the non-deterministic knowledge, and the relationship between the entities in the knowledge graph is inferred so as to accurately and reasonably obtain results. Experimental results from real knowledge graph datasets show that the accuracy of KGIPSL is better than that of existing baseline methods, and the efficiency has also been significantly improved. The main contributions of this paper are as follows:

- This paper proposes a knowledge graph relationship inference method, KGIPSL, based on probabilistic soft logic. KGIPSL uses a knowledge graph based on Markov logic network to build a knowledge base and employs the random walk method to sample relevant knowledge needed in the process of inference.

* Corresponding author.
E-mail address: kitesmile2000@gmail.com

- KGIPSL is beneficial to the deep inference of knowledge graphs and can better infer the relationship between uncertain entities. KGIPSL can mine the potential relationship between entities according to the existing knowledge in the knowledge graph, infer new knowledge, and improve the knowledge coverage and completeness of knowledge graphs.
- We conducted a large number of experiments on real datasets. The results of a large number of experiments show that KGIPSL outperforms the existing baseline methods in terms of accuracy, recall, and efficiency, and the average accuracy of KGIPSL on the YAGO dataset is 14.9% higher than that of the baseline method.

The rest of the paper is organized as follows: Section 2 summarizes the related work, Section 3 gives a detailed description of the knowledge graph relationship inference problem, and Section 4 presents the proposed probabilistic soft logic-based knowledge graph relationship inference framework. The experimental results and analysis are detailed in Section 5, and the paper is summarized in Section 6.

2. Related Work

Traditional methods of knowledge inference are mainly divided into inductive inference, deductive inference, and representation-based inference. Inductive inference mainly learns inference rules. Typical methods include inductive logic programming, association rule mining, and path sorting algorithms. Inductive logic programming (ILP) [2] focuses on learning relational knowledge (such as logic programs) from examples. Although such methods are very effective in the rule discovery of small datasets, finding the Horn clause rules in a series of facts is very time consuming [3]. There is also insufficient support for real-time inference, and the accuracy of sparse data is not high enough to be extended to in a large knowledge base. AMIE mines closed rules from incomplete knowledge bases to learn the rules for predicting relationships between entities [4]. For each relationship, starting from the rule that the rule body is empty, the rule body part is expanded by three operations, and the candidate rule whose support degree is greater than the threshold is retained. The idea of the path sorting algorithm is that there is a correspondence between the rules in the knowledge graph and the relationship between the entities [5-6]. The PRA uses the path between entities as a feature to learn the classifier of the target relationship. It first uses the method of random walk, breadth-first search, or depth-first search to extract features, and then it performs feature calculation according to the probability of traverse and the frequency of occurrence. Finally, joint learning is performed on various relationships to train the classifier. Deductive inference is the inference of specific facts, mainly including methods based on Markov logic networks and probabilistic soft logic [7]. The shortcoming of the inference method based on probabilistic graphical model is that the interpretability is weak. Probabilistic soft logic is an extension of the Markov logic network, and it allows the true value of facts to be arbitrarily selected in the continuous interval [0, 1], which simplifies the discretization problem into continuous optimization problems, improves the efficiency of inference, and enhances the ability of Markov logic networks to deal with uncertainty.

The core idea of knowledge-based inference methods based on representation learning is to represent symbolized entities and relationships in a continuous vector space, which preserves the original graph structure while simplifying the operations and calculations [8-10]. Such methods require constructing a vector space model of entities and predicates according to knowledge base, defining scoring functions, evaluating the probability that each triple is true, and inferring new facts according to the probability. Such methods are simple and easy to calculate, but they cannot use logical information to introduce implicit knowledge, and the inference process lacks interpretability. Rescal, which is based on the sparse tensor method, learns relationships from semantic web association data [8]. It can be extended to millions of levels of entities, relationships, and facts. Rescal also incorporates ontology knowledge in the decomposition process to improve learning and distributed parallel computing on multiple nodes. However, it still cannot alleviate the high memory overhead and low operating efficiency inherent in the tensor decomposition model. TransE, a pioneer in the embedding method, mainly solves the problem of embedding entities and relationships in low-dimensional vector space in multidimensional relational data [9]. TransE is easy to train, has fewer parameters, and is easily extended to large-scale knowledge graphs. This method models the entity relationship through the translation relationship of the low-dimensional entity embedded space, which represents each prediction as a translation vector from subject to object. Nickel et al. proposed a method of embedding (HolE) to learn the combined vector space representation on the entire knowledge graph [10]. This method is related to the full-featured model of associative memory and uses a circular association method to create a combined vector space representation.

Data from multiple data sources contains a large amount of noise, and information extraction is very difficult. There are many errors and inconsistencies in the process of building knowledge graphs. The way in which the knowledge graph is constructed is the first challenge of knowledge graph inference. Much of the knowledge extracted from the real world is non-deterministic and inconsistent. Accurately inferring with this kind of uncertainty knowledge is a challenge. Another

challenge is effectively sampling in the inference process to make inference more efficient. Therefore, in order to meet the requirements of accuracy and efficiency of large-scale knowledge graph inference, it is necessary to combine knowledge based on probability and logic. Take Figure 1 as an example: Tom and Mike are classmates, Tom graduated from university C, and the fact that Mike graduated from university C can then be inferred. In contrast, the vector space model approach finds that the vector representations of Tom and Mike are similar, so the word vectors of the universities they graduated will have similar values. The method of logical inference is more accurate and more inferential than the vector space model, but it is more fragile and slower when it encounters large-scale predicates.

This article utilizes knowledge to build relationships between entities based on the Markov logic network to address data inconsistencies caused by noise data, employs the probabilistic logic soft approach to construct knowledge graphs, and improves the accuracy of inference in the process of reason. The sampling method of breadth-first search is utilized to improve the efficiency of inference. In the specific implementation scheme, the comprehensive consideration of various constraints requires joint inference of the extracted knowledge and the use of probabilistic soft logic to solve the learning and inference process.

3. Problem Description

A knowledge graph is a set of entities and relations between entities, and it can be generated directly from the knowledge triple. Each entity in the triple in the knowledge graph corresponds to a node in the graph (e.g. $N = \{e_s|(e_s, r, e_t) \in T\} \cup \{e_t|(e_s, r, e_t) \in T\}$). The type of edge corresponds to the relationship type in the triple ($R = \{r| (e_s, r, e_t) \in T\}$). Finally, each triple in the knowledge base creates an edge type r of the entity between nodes e_s and e_t. The knowledge base is represented by a tensor K of rank 3. This is a binary tensor (value 0 or 1), where the first tensor value corresponds to the source entity in the knowledge base, the second corresponds to the target entity, and the third corresponds to the relationship type. K_{ijk} is 1 if and only if $(e_i, r_k, e_{j)}$ is a tuple in the knowledge base, and e_i and e_j are entities in the tuple.

Figure 1 shows an example of the problem of knowledge graph inference, where the solid lines represent the relationships between entities and the dashed lines represent the relationships that may exist between entities. From the figure, we can observe the following facts: Tom and Mike worked at the same IT company D, they are John's students, Mike lives in city B, B is the capital of country A, company D is located in country A, Mike graduated from university C, and his major was computer science. It is presumed that Tom and Mike are classmates, and Tom also studied computer science at university C. Thus, we can take advantage of the redundant information knowledge base of relationships between entities and speculate the latent knowledge. The relationships in the graph can be described with the following triples:

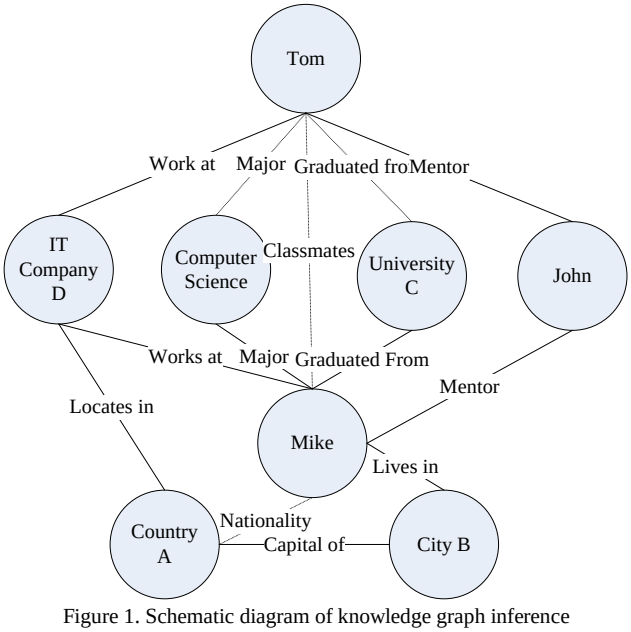


Figure 1. Schematic diagram of knowledge graph inference

<Tom, work at, IT company>
<Mike, work at, IT company>
<Jorn, mentor, Tom>

<Jorn, mentor, Mike>
< Mike, graduated from, University C>
<Mike, lives in, City B>
<City B, capital of, Country A>
<IT company, locates in, Country A>

Find < Tom, major, ?>, <Tom, graduated from, ?>, <Tom, ?, Mike>, <Mike, nationality, ?> from these rules, and reason the relationship between these relationships and entities. Probability is a problem that knowledge inference must solve. The objective of this paper is to minimize the distance between the predicted tuple and the true tuple in semantic distance. The following objective function can be used to minimize the semantic distance between tuples:

$$\min_{\theta} \sum_{x^+ \in D} \sum_{x^- \in D^-} L(f(x^+;\theta)) + \lambda reg(\theta) \tag{1}$$

Where θ is all the parameters in the model, $reg(\theta)$ is the normalized function of θ, W_k is the weighted matrix, x^+ is the tuple in knowledge base D, x^- is the new tuple predicted in the predicted result set D^-, λ is the normalized coefficient, and $L(f, f')$ is a marginal loss function.

$$L(f, f') = \max(1 + f' - f, 0) \tag{2}$$

In this paper, the latent semantic distance model is used to measure the semantic distance between tuples, and the objective function is optimized by stochastic gradient descent. The convergence time is reduced by the alternating least squares method. The proposed method has the following advantages: first, it does not assume a negative sample value 0, but it is less positive than the positive samples. Second, f can be any function of the form of the function; the probability function is not limited. It is assumed that the larger the f value, the greater the possibility that the tuple is true. The objective function is easily optimized by random gradient descent. In each iteration, there is only a positive sample and a negative sample. This method is also scalable on large-scale data. However, its convergence time is longer. As mentioned earlier, some models can be optimized by alternating least squares when used with the squared loss objective function, thereby reducing the convergence time.

4. Proposed Method

In this section, the details of the proposed method are given in detail. Firstly, the relationship between entities in the knowledge graph is constructed based on the Markov logic network. Then, the facts in the knowledge graph are sampled based on random walk. Finally, the probabilistic soft logic method for knowledge inference is utilized for sampled facts.

4.1. Sorting of Entity Relationships based on Statistics Method

Bhatia et al. proposed a relationship sorting algorithm to find the target entity corresponding entities and their relationships [11]. Because each relationship mapping knowledge in the number of occurrences in the knowledge base library can be statistically derived, the text referred to and the number of relations in the knowledge graph indicate the strength of support or relations. This method gives the entity e_s and the relation set Re_s (R$e_s \subseteq$ R) containing e_s, which can be derived from the relationship list Re_s sorted according to importance or relevance. Two steps are needed to obtain the sorted list: first, KGIPSL selects the target entity e_t according to the input entity e_s, and then it selects an edge r connecting the two entities. For example, for an input entity, first select the target entity, and then determine the selection method of the relationship between the two entities. Since there may be multiple relationships between the two entities, it is necessary to sort the relationships to find the closest relationships. This paper uses the following formula to rank the two relationships:

$$P(r, e_t \mid e_s) = P(e_t \mid e_s) P(r \mid e_t, e_s) = P(r \mid e_s, e_t) \frac{P(e_t) P(e_s \mid e_t)}{P(e_s)} \tag{3}$$

Where $P(e_s)$ is the same for all entities and relationships, so it can be ignored, and then the above formula can be rewritten as

$$P(r, e_t \mid e_s) \propto P(e_t) \times P(e_s \mid e_t) \times P(r \mid e_s, e_t) \tag{4}$$

The ranking function of the above formula can sort all the relationships of a given entity. This function contains three parts: entity priori, entity similarity, and relationship strength. Among them, the entity prior is that, in the absence of other external information, the probability that an entity has a relationship with a popular entity in the knowledge graph is higher than the probability of a relationship with a rare entity.

$$P(e_t) \propto relCount(e_t) \tag{5}$$

The entity class considers that the more relationships between a target entity and the input entity, the closer the relationship between the two entities, and the relationship between the target entity and the input entity accounts for the relationship between the target entity and all entities. The more relationships between the target entity and the input entity, the higher probability they belong to the same entity class. For example, entity A has almost the same number of relationships with entities B and C, and the relationship between entities A and C owns a large proportion of all the relationships connected to entity C. Therefore, compared with entity B, entity A is more similar to entity C. Use Equation (6) to represent the similarity between relationships.

$$P(e_s \mid e_t) = \frac{\sum_{r_i \in R(e_s, e_t)} w(r_i)}{\sum_{r_i \in R(e_t)} w(r_i)} \tag{6}$$

Where $R(e_t)$ is the set of relationships from entities e_t, $R(e_s, e_t)$ is the set of all relationships between entities e_s and e_t, and $w(r_i)$ is the weight of the relationship between r_i. The entity priori and the entity class mainly focus on the relative importance of the target entity to the input entity, and the relationship strength indicates the importance of the different relationship types corresponding to the target entity. Since there may be multiple relationships between two entities, in the absence of other ancillary information, it is assumed that the relationship with more support in the knowledge graph is more important than the relationship with less support. The strength of the relationship is given by

$$P(e_s \mid e_t) = \frac{mentionCount(r, e_s, e_t)}{\sum_{r \in R(e_s, e_t)} mentionCount(r, e_s, e_t)} \tag{7}$$

Where $mentionCount(r, e_s, e_t)$ represents the number of relationships r of entities e_s and e_t in the knowledge graph.

4.2. Construction Method based on Conditional Random Field Model

In traditional relational data model, logic-based knowledge is the possible knowledge of the world along with a set of hard limits. However, in the real world, many are non-deterministic. We can thus add a soft limit such that if the fact breaches a rule, it will not disappear, which will reduce the possibility of its existence. In the relational data, data dependency exists between entities, and each tuple can depend on any knowledge graph random variable due to the large number of possible dependencies. Even for small-scale knowledge graphs without further restriction, the joint probability distribution may quickly fall into the solution dilemma. In order to reduce the number of possible dependencies and find a solvable method, we design a template-based graph model to reason the dependencies between tuples.

The graphical model encodes the dependence between the random variables with probabilistic graphs. Each random variable y_{ijk} is a node in the graph, and each dependency corresponds to an edge in the graph. In order to distinguish such a graph from a knowledge graph, this graph is called a dependency graph. The key difference between knowledge graphs and dependency graphs is that knowledge graphs code existing facts, while dependency graphs code the statistical dependence between facts (random variables). To avoid the problem of cyclic dependencies, we use Markov random fields (MRFs) to model dependency graphs, as described in Equation (8).

$$P(\underline{Y} \mid \theta) = \frac{1}{Z} \prod_c \psi(y_c \mid \theta) \tag{8}$$

Where $\psi(y_c \mid \theta) \geq 0$, which is a possible function of a variable in the c^{th} subset corresponding to the c^{th} group on the dependence graph, and $Z = \sum_y \prod_c \psi(y_c \mid \theta)$ guarantees a division function whose sum of distributions is 1. It is possible that the function captures the local relationship between the variables in each group c in the graph. Note that unlike the directed graph model, there is no probability interpretation of local potential in the undirected graph model. This formula

again defines the probability distribution of the possible world, such as the joint probability assignment of the random variable *Y*.

The structure of the dependency graph can be generated from the template mechanism, and there are many ways to generate templates. A common method is to use Markov logic, which is a template language based on logic rules [12]. Given a set of rules $F = \{F_i\}_{i=1}^{L}$, if a fact appears in a benchmark rule set, an edge can be created between the nodes that depend on the graph. The inference problem involves estimating the most probable configuration, $y^* = argmax P(y \mid \theta)$, or the posterior marginal probability $P(y_i|\theta)$. In short, both of these problems are difficult to calculate. If the limiting potential function is only the disjunction function, a special Markov random field with hinge loss can be obtained, which is a convex optimization problem that can be efficiently based on the relaxation of continuous binary random variables [13]. Since the probability of soft logic can be described as non-deterministic HL-MRFs facts, this paper uses probabilistic soft logic for knowledge graph inference.

4.3. Local Closed World Assumption

A closed world assumption approach is used to obtain a counterexample in the traditional basis of inductive logic programming problems [14]. It declares all unseen positive samples as counter-examples. However, because knowledge graphs miss a large amount of data, this assumption is not practical. Although knowledge graphs contain many facts, using the closed world assumption method will generate several negative samples, and a negative sample training the model can be misleading. The knowledge graph predicate represents almost all binary relations. It does not mean that the predicate counter-example, which assumes that samples that are not positive examples of statements can be counter-examples, may be an unknown category. The open world assumption states that knowledge graphs are considered to not be negative samples, but rather unknown facts [15]. Knowledge graphs contain only positive training examples, so they do not encode the facts wrong. We utilize the local closed world assumption (LCWA) to obtain negative samples; that is, if the knowledge graph cannot determine whether the fact is true, negation facts will be used as negative samples [16]. We assume knowledge graphs are partially complete, and the negative samples are obtained using LCWA. In the process of calculating the node feature (source node and destination node), if it is not a positive sample, then it is considered to be a negative sample.

4.4. Sampling Method based on Random Walk

The Metropolis random walk method is a Markov chain Monte Carlo (MCMC) sampling method that generates a sequence of random walks and decides whether to accept the sequence by rejecting the sampling method. We present a simulation using the Metropolis Markov chain random walk method, where each state X_{t+1} depends only on its previous state X_t. X_t is given by the current state of a new sample *X'*, which has a probability of α acceptance into the next state X_{t+1}. α is derived from Equation (9), where $P(X)$ is a state probability of *X*.

$$\alpha = P(X_{t+1} = X' \mid X_t) = \min\left\{1, \frac{P(X')}{P(X_t)}\right\} \tag{9}$$

In the knowledge graph, *u* is any node in the probabilistic graph *G*, and $V(u)$ is a set of neighbor nodes of *u*. *W* is a random sample in $V(u)$. $WD^G(i)$ represents the degree of vertex *i* in graph *G*. Equation (9) can be written as Equation (10), where $P(u- > w)$ represents the random walk probability from *u* to *w*. For the random walk in each attempt, the current state of states *u* to *w* is $V(u)$. Thus, the Metropolis random walk begins with the initial vertex and then samples along the edges in the graph *G*. Therefore, the Markov chain of the sampling nodes in the graph *G* can be obtained by randomly traveling the sampler through the Markov chain.

$$P(u \to w) = \min\left\{1, \frac{WD^G(w)}{WD^G(u)}\right\} \tag{10}$$

4.5. PSL-based Knowledge Graph Inference

Since the corresponding dependency graph in the knowledge graph is generated by the random walk method, the relationship between the nodes in the graph is a probability value representation between 0 and 1. A weighted dependency graph is based on the knowledge graph. The weight is the truth probability of the fact between two nodes. Because there are many errors or inconsistent knowledge in the process of constructing the knowledge graph, it is difficult for the method of

deterministic inference to deal with the uncertain facts in the knowledge graph. Probabilistic soft logic combines the advantages of probabilistic graph models and first-order logic, and it is applicable to the inference of knowledge graphs with relational feature data. Therefore, we propose a method based on probabilistic inference to describe the uncertainty of facts in the real world.

Firstly, according to the dependency graph, a template Markov network is constructed. Then, the probability of a soft non-logical inference is used to determine the relationships between the facts and finally fuse the factual conflicts from different data sources. This paper presents a method of inference (knowledge graph based on the probability of soft logic inference based on PSL, which is abbreviated as KGIPSL) for knowledge graph inference, and the BFS method is utilized to reduce the computational complexity and shorten the inference process time. The details of the KGIPSL algorithm are given in Table 1.

Table 1. KGIPSL: Knowledge graph inference algorithm based on probabilistic soft logic

Algorithm on knowledge graph inference based on PSL
Ensure: A Markov chain of vertices in G: $X = X_0, X_1, \cdots, X_{max}$
1: $X_0 = u$;
2: $L = \varnothing$;
3: $L = L \cup X_0$;
4: for ($i = 0$; i++; i<max)
5: {
6: if (X_i is a node in the core subgraph) // The core subgraph deletes the edges and nodes with less weight from G
7: Return;
8: else
9: {
10: X'_{i+1} = adjacentlist(X_i);
11: α = randomwalkrate(); //$\alpha \in [0,1]$;
12: if ($\alpha \le \min\left\{1, \frac{P(X_{i+1})}{P(X_i)}\right\}$)
13: { $X_{i+1} = X'_{i+1}$};
14: else
15: {$X_{i+1} = X_i$};
}
16: $L = L \cup X_{i+1}$;
17: I ++;
18: }

5. Experimental Results and Analysis

Firstly, the KGIPSL algorithm runs on the Alchemy package, which is based on a statistical relational learning package Markov logic and inference [17]. It can be used to resolve entity resolution, link prediction, social network modeling, information extraction, and other issues. The local closed world hypothesis obtains a negative sample. No negative sample is encountered if the sample is considered positive. In the experiment, a fine particle size training data and cross validation of the movable grid search are utilized to adjust the parameters. The parameters of the random walk vector space and the number of steps in the calculation of feature selection in KGIPSL are determined.

The experimental platform uses an Intel Core i7-2640M 2.80 GHz processor, 32G running memory, and Centos6.4 operating system. Depending on whether the tuple reasoned by the knowledge graph is a correct and reasonable tuple, the reasonable meta-component is one class, and the unreasonable meta-component is another class, which translates into a reasonable dichotomy problem of a tuple. A specific relationship similarity threshold is used to identify a tuple is reasonable. If a tuple <*s*, *p*, *o*> calculated by $f_p(s, o)$ similarity is greater than the threshold value, the tuple is predicted as a correct tuple; otherwise, it is an error tuple. KGIPSL's similarity threshold is obtained in the classifier training process. This paper uses the accuracy and recall rate to evaluate the classification effect of the tuple. The accuracy rate is the ratio of the inferred accurate rules to the correct rules. The recall rate is the ratio of the inferred rules to the total rules.

5.1. Datasets

This paper uses the knowledge graph composed of person entities in the YAGO and Freebase datasets to conduct experiments. Because the scale of the Freebase knowledge base is very large, we first filter the relationship that is easy to extract, and some of them have a large-scale relationship. There are 350 M tuples in the filtered dataset. Because most of

them have different relationships with the entities to be predicted, we further filter the tuples in the experiment and only keeps the training and test examples that may be connected in the experiment. The vector space is obtained using the relationship between the latent layers of the tuple represented with principal component analysis.

We select 20 kinds of relations from YAGO and 24 kinds of relations in Freebase to test the proposed model. YAGO relations are selected by hand. The known examples are divided into a training set and test set, which respectively represent 80% and 20% of the total data. Each class relationship contains 810 training examples and 190 test samples on average. The 24 relationships in Freebase are randomly chosen. 4,338 relations are filtered based on the following principle: the number of instances of the relationship is between 1,000 and 10,000, and there is no relationship between the intermediate nodes. Once the relationship is selected, all instances that may be connected to the tuple data are retained, which leaves an average of 200 instances for each relationship. Finally, this data is divided into training and test datasets according to the ratio of 80% to 20%.

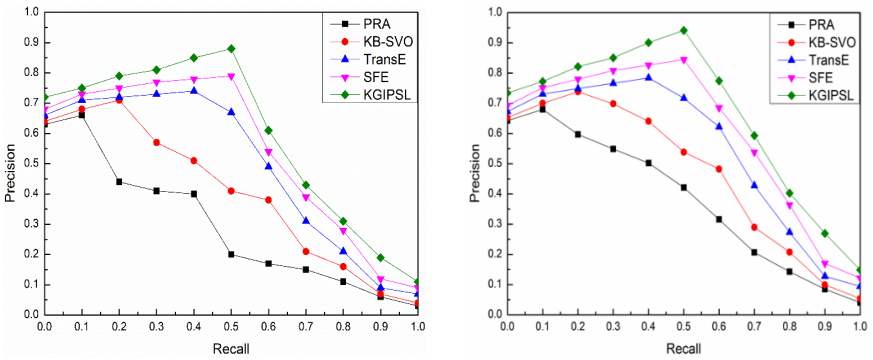
5.2. Baseline Methods

We employ PRA, KB-SVO, TransE, and SFE as baseline methods. PRA is a knowledge inference method based on the path sorting algorithm [18]. This method implements knowledge inference by constraining inference rules as the preconditions of Horn's clause and corresponding to the path in the graph. KB-SVO uses the distributed semantics of entities and symbol logic for knowledge inference and uses the physical similarity vector space to solve the semantic lack problem, which is caused by latent sparse features in the random walk sample process [19]. TransE models the relationship between entities by translating the word embedding vectors of entities into low-dimensional space, and it uses the similar relationships between entity word embedding vectors for knowledge graph inference [9]. Finally, SFE uses the breadth-first search method to find the subgraphs in the vicinity of the entity pair and extracts the subgraph structure features, inferring knowledge according to the relationships between the extracted subgraph structure feature entities [6].

5.3. Experimental Results and Analysis

Figure 2(a) shows the precision and recall curve results of the baseline methods and KGIPSL on the YAGO dataset. It can be seen from the figure that KGIPSL is superior to the baseline methods; the accuracy is 0.88 when the recall is 0.5, which is 0.09 and 0.11 higher than SFE and KB-SVO, respectively. The average accuracy of KGIPSL is 14.9% higher than those of the baseline methods on average when the recall rate increases from 0.1 to 1.

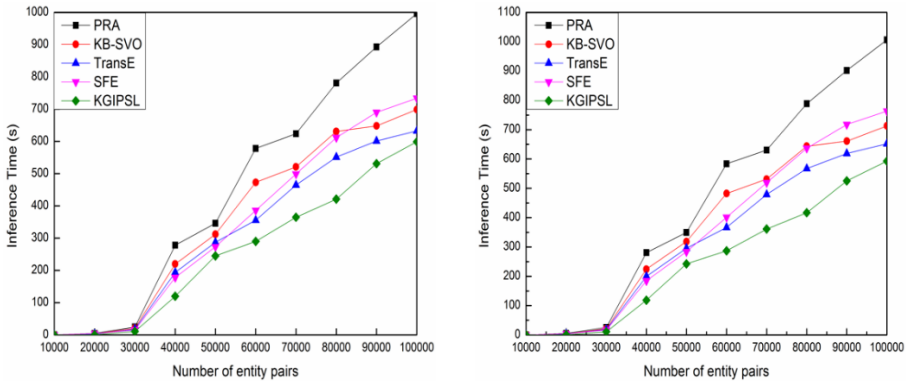
Figure 2(b) shows the results of accuracy and recall rate of the baseline methods and KGIPSL on the Freebase dataset. It can be seen from the figure that KGIPSL performs significantly better than the other methods; the accuracy is 0.94 when the recall rate is 0.5, which is 0.1 higher than the following SFE method, and the average is 0.38 higher than the averages of the baseline methods. The AUC of KGIPSL is 0.68, which is 0.06 higher than the following SFE method.



(a) YAGO dataset (b) Freebase dataset

Figure 2. Accuracy recall curve on YAGO and Freebase datasets

Figure 3(a) shows the comparison result of KGIPSL to baseline methods of inference time on the YAGO dataset. It can be seen from the figure that KGIPSL consumes the shortest amount of run time among the baseline methods. For example, the inference time of the KGIPSL method is 421 seconds on 80,000 entity pairs; the minimum of the baseline method is TransE, which consumes 551 seconds; and the maximum run time, 781 seconds, is observed with the PRA method. The run time of KGIPSL is 223 seconds less than those of the baseline methods on average.



(a) YAGO dataset (b) Freebase dataset

Figure 3. Comparison results of knowledge graph inference time

Figure 3(b) shows the comparison results of different methods of inference time on the Freebase dataset. It can be seen from the picture that KGIPSL consumes the minimum run time among the baseline methods. For example, the run time of KGIPSL is 526 seconds on 90,000 entity pairs, while TransE and PRA consume 619 and 902 seconds, respectively. The run time of KGIPSL is 191 seconds less than those of the baseline methods on average.

6. Conclusions

Knowledge graph inference is the key problem of knowledge graph construction. We propose a knowledge graph inference method KGIPSL based on a probability logic soft approach. KGIPSL first uses knowledge to build a knowledge-based graph based on the Markov logical network, then utilizes the random walk method to sample the required knowledge, and finally employs probabilistic soft logic for knowledge graph inference. Extensive experimental results show that KGIPSL is superior to the existing baseline methods in terms of accuracy, recall rate, and efficiency. The average accuracy of KGIPSL on the YAGO dataset is 14.9% higher than that of the baseline method. The inference method proposed in this paper is beneficial to the deep inference of knowledge graphs and can reasonably infer the relationships between uncertain entities. KGIPSL can mine the potential relationship between entities according to the existing knowledge in the knowledge graph, infer new knowledge, and improve the knowledge coverage and completeness of knowledge graphs. The knowledge graph inference process encounters some knowledge conflicts, which are caused by the inconsistency of knowledge graphs evolving over time. We will explore the knowledge confliction problem to improve knowledge graph quality in our future work.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No. 61672470,61702462, 61802352, 61866008), Doctoral Research Fund of Zhengzhou University of Light Industry (No. 2017BSJJ046, 2018BSJJ039, 13501050045), Science and Technology Project of Henan Province (No. 182102210617, 182102210607, 2017BSJJ046), National Key Research and Development Plant (No. 12016YFE0100600, 12016YFE0100300), Second Education Fund for Industry and Education Project "Digital Science and Technology, Wisdom for the Future" (No. 2018A01094), Henan Province Educational Committee (No. 17A520064), and CERNET Innovation Project (No. NGII20161202).

References

1. R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin, "Knowledge Base Completion via Search-based Question Answering," in *Proceedings of the 23rd International Conference on World Wide Web* (*WWW*), pp. 515-526, Seoul, Korea, April 2014
2. S. Muggleton and L. D. Raedt, "Inductive Logic Programming: Theory and Methods," *Journal of Logic Programming*, Vol. 19, pp. 629-679, May 1994
3. J. R. Quinlan, "Learning Logical Definitions from Relations," *Machine Learning*, Vol. 5, No. 3, pp. 239–266, August 1990
4. L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, "AMIE: Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases," in *Proceedings of the 22nd International Conference on World Wide Web* (*WWW*), pp. 413-422, Rio de Janeiro, Brazil, May 2013
5. N. Lao, T. Mitchell, and W. W. Cohen, "Random Walk Inference and Learning in a Large Scale Knowledge Base," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pp. 529-539, Edinburgh, United Kingdom, July 2011
6. M. Gardner and T. M. Mitchell, "Efficient and Expressive Knowledge Base Completion using Subgraph Feature Extraction," in

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (*EMNLP*), pp. 1488-1498, Lisbon, Portugal, September 2015

7. A. Kimmig, S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "A Short Introduction to Probabilistic Soft Logic," in *Proceedings of the NIPS Workshop on Probabilistic Programming*: *Foundations and Applications*, pp. 1-4, Nevada, United States, December 2012
8. M. Nickel, V. Tresp, and H. -P. Kriegel, "Factorizing Yago: Scalable Machine Learning for Linked Data," in *Proceedings of the 21st International Conference on World Wide Web* (*WWW*), pp. 271-280, Lyon, France, April 2012
9. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-Relational Data," *Advances in Neural Information Processing Systems 26*, pp. 2787-2795, Nevada, United States, December 2013
10. M. Nickel, L. Rosasco, and T. Poggio, "Holographic Embeddings of Knowledge Graphs," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1955-1961, Phoenix, United States, February 2016
11. S. Bhatia, A. Goel, and A. Jain, "Separating Wheat from the Chaff – A Relationship Ranking Algorithm," in *Proceedings of European Semantic Web Conference* (*ESWC*), pp. 79-83, Heraklion, Crete, Greece, May 2016
12. O. Kuželka and J. Davis, "Markov Logic Networks for Knowledge Base Completion: A Theoretical Analysis under the MCAR Assumption," in *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 427-437,Tel Aviv, Israel, July, 2019
13. S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "Hinge-Loss Markov Random Fields and Probabilistic Soft Logic," *Journal of Machine Learning Research*, Vol. 18, pp. 109:1-109:67, 2017
14. J. Minker, "On Indefinite Databases and the Closed World Assumption," in *Proceedings of the 6th Conference on Automated Deduction*, pp. 292-308, New York, United States, June 1982
15. N. Drummond and R. Shearer, "The Open World Assumption," *eSI Workshop*: *The Closed World of Databases Meets the Open World of the Semantic Web*, Vol. 15, pp. 1-23, October 2006
16. X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, et al., "Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 601-610, New York, United States, August 2014
17. A. Nath and M. Richardson, "Counting-MLNs: Learning Relational Structure for Decision Making," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 1068-1074, Toronto, Canada, July 2012
18. N. Lao and W. W. Cohen, "Relational Retrieval using a Combination of Path-Constrained Random Walks," *Machine Learning*, Vol. 81, No. 1, pp. 53-67, October 2010
19. M. Gardner, P. P. Talukdar, J. Krishnamurthy, and T. Mitchell, "Incorporating Vector Space Similarity in Random Walk Inference over Knowledge Bases," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pp. 397-406, Doha, Qatar, October 2014