Knowledge Graph Embedding via Relation Paths and Dynamic Mapping Matrix

Shengwu Xiong[1,2], Weitao Huang[1], and Pengfei Duan[1,2](✉)

[1] School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China
{swxiong,duanpf}@whut.edu.cn
[2] Hubei Key Laboratory of Transportation Internet of Things, Wuhan 430070, China

Abstract. Knowledge graph embedding aims to embed both entities and relations into a low-dimensional space. Most existing methods of representation learning consider direct relations and some of them consider multiple-step relation paths. Although those methods achieve state-of-the-art performance, they are far from complete. In this paper, a noval path-augmented TransD (PTransD) model is proposed to improve the accuracy of knowledge graph embedding. This model uses two vectors to represent entities and relations. One of them represents the meaning of a(n) entity (relation), the other one is used to construct the dynamic mapping matrix. The PTransD model considers relation paths as translation between entities for representation learning. Experimental results on public dataset show that PTransD achieves significant and consistent improvements on knowledge graph completion.

Keywords: Representation learning · Knowledge graph
Dynamic mapping matrix · Relation path

1 Introduction

Knowledge graphs encode structured information of entities and their rich relations. Although typical knowledge graphs, such as Freebase [18] (Bollacker et al. 2008), WordNet [19] (Miller 1995), Yago [20], usually are large in size and contain thousands of relation types, millions of entities and billions of triples, they are usually far from complete. Knowledge graphs usually contain large-scale structural information by using the form of triples (head entity, relation, tail entity).

Knowledge graph completion is similar to link prediction in social network analysis [11], but traditional approach of link prediction is not capable for knowledge graph completion. Inspired by the translation invariant phenomenon in word vector space [10], Bordes et al. propose the TransE [3] model, which achieves state-of-the-art prediction performance. TransE learns low-dimensional embeddings for each entity and relation. These vector embeddings are donated by the

C. Woo et al. (Eds.): ER 2018 Workshops, LNCS 11158, pp. 106–118, 2018.
https://doi.org/10.1007/978-3-030-01391-2_18

same letter in boldface. The basic idea of TransE is that every relation is regarded as translation between head entity and tail entity in the embedding space. For example, for a triplet (h, r, t), the embedding h is close to the embedding t by adding the embedding r, that is $h + r \approx t$. TransE has outstanding performance in 1-to-1 relations, but it's not good at dealing with 1-to-N, N-to-1 and N-to-N.

In order to overcome the problems of TransE [3] in modeling 1-to-N, N-to-1, N-to-N relations, Researchers propose many methods such as TransH [13], TransR [7] to extend the model and achieve excellent performance. These methods are proposed to assign an entity with different representations when involved in various relations. TransH enables an entity to have distributed representations when involved in different relations. However, TransE and TransH both assume embeddings of entities and relations being in the same space R^k is the semantic space dimension, so it's hard for TransE, TransH to distinguish among the entities which are on the many sides. To deal with the above problem, TransR is proposed to model entities and relations in distinct space, i.e., entity space and multiple relation spaces [4], and performs translation in the corresponding relation space. TransR achieves significant improvements compared to base-lines including TransE and TransH.

In both TransH [13] and TransR [7], all types of entities share the same mapping vectors/matrices. However, different types of entities have different attributes and functions, it is insufficient to let them share the same transform parameters of a relation [14]. And for a given relation, similar entities should have similar mapping matrices and otherwise for dissimilar entities. Furthermore, the mapping process is a transaction between entities and relations that both have various types. TransD [9] names two vectors to represent symbol object (entities and relations). The first one is used to construct mapping matrices, the other one captures the meaning of entity (relation). TransD has less complexity and more flexibility than TransR/CTransR. When learning embeddings of named symbol objects (entities or relations), TransD considers the diversity of them both.

One shortcoming of the above approaches is that they only consider the direct relation between entities. In a knowledge graph, it's known that there are also substantial multiple-step relation paths between entities indicating their semantic relationships. For example, the relation path $h \stackrel{son}{\rightarrow} e_1 \stackrel{son}{\rightarrow} t$ indicates the relation grandson between h and t, i.e., $(h, grandson, t)$. Fortunately, the problem is already alleviated effectively by using information of relation paths. PTransE (Guu et al., 2015) [1] tries to extend TransE to model relation paths for representation learning of knowledge graphs and achieves significant and consistent improvements on knowledge graph completion and relation extraction from text.

In this paper, we aim at enhancing TransD [9] with information of relation paths and finally propose a new variant named PTransD. The model is evaluated on a typical large-scale knowledge graph Freebase (Bollacker et al., 2008). In this paper, we adopt a dataset extracted from Freebase, i.e., FB15K. Experimental results show that modeling relation paths provide a good supplement for representation learning of knowledge graphs.

2 Our Method

In Sect. 1, we introduce the advantages and disadvantages of existing methods, including TransE [3], TansH [13] and TransR/CTransR [7]. In this section, PTransD model will be introduced in detail.

2.1 Based Method: TransD

Model TransR/CTransR [7] segments triples of a specific relation r into several groups and learns a vector representation for each group. However, entities also have various properties. In both TransH [13] and TransR/CTransR [7], all types of entities share the same mapping vectors/matrices. It is insufficient to show different attributes and functions of different types of entities. And for a given relation similar entities should have similar mapping matrices and otherwise for dissimilar entities. TransD [9] considers different types of both entities and relations, to encode knowledge graphs into embedding vectors via dynamic mapping matrices produced by projection vectors.

In TransD [9] model, each named symbol object (entities and relations) is represented by two vectors. The first one captures the meaning of entities (relations), the other one is used to construct mapping matrices. For example, given a triplet (h, r, t), its vector are h, h_p, r, r_p, t, t_p, where subscript p marks the projection vectors, $h, h_p, t, t_p \in R^n$ and r, $r_p \in R^m$. For each triplet (h, r, t), TransD [9] set two mapping matrices M_{rh}, $M_{rt} \in R^{m \times n}$ to project entities from entity space to relation space. They are defined as fellow:

$$M_{rh} = r_p h_p^\top + I^{m \times n} \tag{1}$$

$$M_{rt} = r_p t_p^\top + I^{m \times n} \tag{2}$$

where $I^{m \times n}$ is an identity matrix of $m \times n$, $h_p^\top$ and $t_p^\top$ are the transpose matrices of h_p and t_p. The mapping matrices M_{rh} and M_{rt} are determined by both entities and relations, and this kind of operation makes the two projection vectors interact sufficiently because each element of them can meet every entry comes from another vector. With the mapping matrices, the projection vectors were defined as follows:

$$h_\perp = M_{rh} h, t_\perp = M_{rt} t \tag{3}$$

The score function is:

$$f_r\left(h, t\right) = -||h_\perp + r - t_\perp||_2^2 \tag{4}$$

In experiments, TransD [9] enforce constrains as $||h||_2 \leqslant 1, ||t||_2 \leqslant 1, ||r||_2 \leqslant 1, ||h_\perp||_2 \leqslant 1, ||t_\perp||_2 \leqslant 1$. To train the model, TransD [9] generate negative samples and use the margin-based rank loss. The overall function of the TransD model is:

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S^-} [\gamma + f_r(h', r', t') - f_r(h, r, t)]_+ \tag{5}$$

where $[x]_+ = max(0, x)$, S and S^- denote golden triples and negative triples. S^- is constructed as follows:

$$S^- = (h', r, t) \cup (h, r', t) \cup (h, r, t') \tag{6}$$

And γ is the margin separating golden triples and negative triples. (h, r, t) and (h', r', t') denote a golden triplet and a corresponding negative triplet, respectively. The process of minimizing the above objective is carried out with stochastic gradient descent (SGD) [12]. In order to speed up the entity and relation embeddings with the results of TransE [3] and initiate all the transfer matrices with identity matrices.

2.2 Path-Based TransD: PTransD

Compared to TransE [3], PTransE [1] achieves consistent and significant improvements on knowledge graph completion and relation extraction from text. Obviously, it is effectively to improve the ability of knowledge representation by considering multiple-step relation paths. Therefore, we propose a path-based TransD [9] model (denoted as PTransD) (Fig. 1).

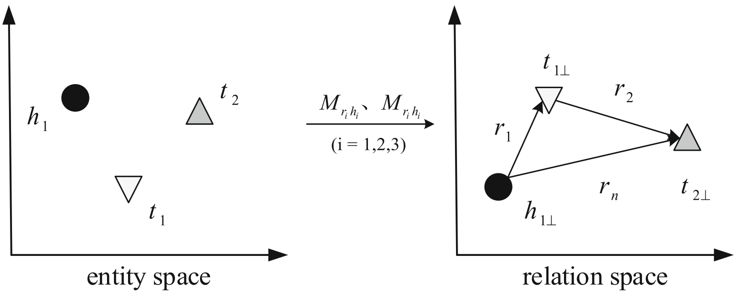

Fig. 1. An illustration of the PTransD model. $M_{r_i h_i}$ and $M_{r_i t_i}$ are mapping matrices of i-th (i = 1, 2, ..., n) head entity h_i and tail entity t_i, PTransD set two mapping matrices to project entities from entity space to relation space.

Multi-step relation paths between entities indicating their semantic relationships. The relation paths reflect complicated inference patterns among in knowledge graphs. However, on the one hand, not all relation paths are meaningful and reliable for learning. For example, a typical relation path $h \xrightarrow{Friend} e_1 \xrightarrow{profession} t$, but actually it does not indicate any semantic relationship between h and t. In experiments, it will bring negative influence. Therefore, in the model we use path-constraint resource allocation algorithm to measure the reliability of relation paths and we select the reliable relation paths for representation learning. To evaluate the reliability of a path, we use a path-constraint resource algorithm (PCRA) just like PTransE [1] model. The basic idea is that assuming a certain amount of resource is associated with the head entity h, and will flow following the given path p. We use the resource amount that eventually flows to the tail entity t to measure the reliability of the path p.

Formally, for a path triple (h, r, t), we compute the resource amount flowing from h to t given the path $p = (r_1, ..., r_l)$. Starting from h and following the relation path p, the following path can be represented by $S_0 \xrightarrow{r_1} S_1 \xrightarrow{r_2} ... \xrightarrow{r_l} S_l$, where $h = S_0$ and $t \in S_l$. For any entity m of $S_i(i.e., m \in S_i)$, its direct predecessors along relation r_i in S_{i-1} is donated as $S_{i-1}(\cdot, m)$, The resource flowing to m is defined as (Fig. 2)

$$R_p(m) = \sum_{n \in S_{i-1}(\cdot, m)} \frac{1}{|S_i(n, \cdot)|} R_p(n) \tag{7}$$

where $S_i(n, \cdot)$ is the direct successors of $n \in S_{i-1}$ following the relation r_i, and $R_p(n)$ is the resource obtained from the entity n. For each relation path p, we set the initial resource in h as $R_p(h) = 1$. By performing resource allocation recursively from h through the path p, the tail entity t eventually obtains the resource $R_p(t)$ which indicates how much information of the head entity h can be well translated. We use $R_p(t)$ to measure the reliability of the path p given (h, t), i.e., $R(p|h, t) = R_p(t)$. It can be simply described as Fig. 3.

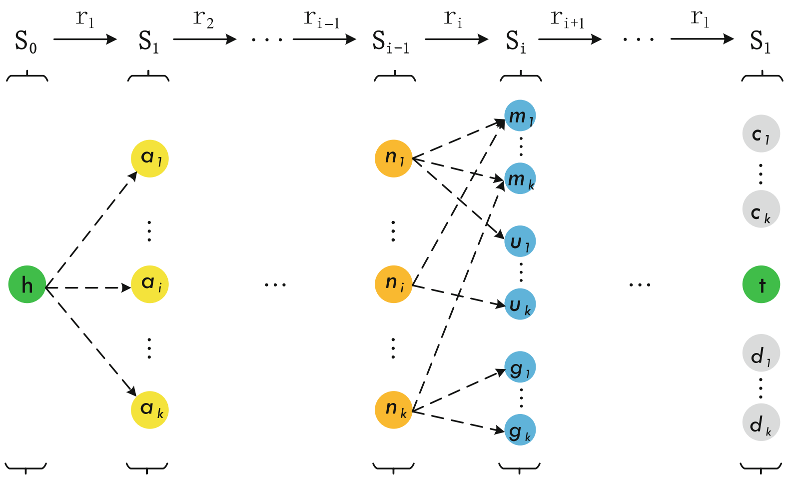

Fig. 2. An illustration of PCRA [17].

On the other hand, it is necessary to represent relation paths in the low-dimensional space. It is obvious that the semantic meaning of a relation path depends on all relations in this path. Given a relation path $p = (r_1, ..., r_n)$, we will define and learn a binary operation function ($\circ$) to obtain the path embedding p by recursively composing multiple relations, i.e. $p = (r_1 \circ ... \circ r_n)$. We add the

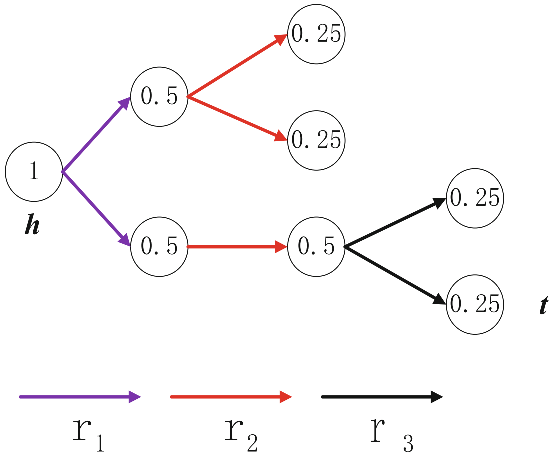

Fig. 3. An simple illustration of PCRA.

embedding of these primitive relations, given by

$$p = r_1 + r_2 + ... + r_n \tag{8}$$

For that in PTransE [1] model, addition operation outperforms other composition operations including multiplication operation and recurrent neural network.

In PTransD model, for each triple (h, r, t), we define the energy function as

$$G(h, r, t) = E(h, r, t) + E(h, P, t) \tag{9}$$

where $E(h, r, t)$ models correlations between relations and entities with direct relations. $E(h, P, t)$ models the inference correlations between relations with multiple-step relation path triples, which is defined as

$$E(h, P, t) = \frac{1}{Z} \sum_{p \in P(h,t)} R(p|h, t) E(h, p, t) \tag{10}$$

where $R(p|h, t)$ indicates the reliability of the relation path p given the entity pair (h, t), $Z = \sum_{p \in P(h,t)} R(p|h, t)$ is a normalization factor, and $E(h, p, t)$ is the energy function of triple (h, p, t).

For a multiple-step relation path triple (h, r, t), we could have followed TransE [3] and define the energy function as $E(h, p, t) = ||h + p - t||$. However since we have minimized $||h + r - t||$ with the direct relation triple (h, r, t) to make sure $r \approx t - h$, we may directly define the energy function of (h, p, t) as

$$E(h, p, t) = ||p - (t - h)|| = ||p - r|| = E(p, r) \tag{11}$$

which is expected to be a low score where the multiple-step relation path p is consistent with the direct relation r, and high otherwise, without using entity embeddings.

In conclusion, we formalize the optimization objective of PTransD as

$$L(S) = \sum_{(h,r,t)\in S} [L(h,r,t) + \frac{1}{Z} \sum_{p\in P(h,t)} R(p|h,t)L(p,r)] \tag{12}$$

Following TransE [3] model, $L(h,r,t)$ and $L(p,r)$ are margin-based loss functions with respect to the triple (h,r,t) and the pair (p,r):

$$L(h,r,t) = \sum_{(h',r',t')\in S^-} [\gamma + E(h,r,t) - E(h',r',t')]_+ \tag{13}$$

and

$$L(p,r) = \sum_{(h',r',t')\in s^-} [\gamma + E(p,r) - E(p,r')]_+ \tag{14}$$

where $[x]_+ = max(0,x)$ returns the maximum between 0 and x, γ is the margin, S is the set of valid triples existing in a knowledge graph and S^- is the set of invalid triples. The objective will favor lower scores for valid triples as compared with invalid triples. The invalid triples set with respect to *(h,r,t)* is defined as $S^- = (h',r,t) \cup (h,r',t) \cup (h,r,t')$. That is, the set of invalid triples is compose of the original valid triple (h,r,t) with one of three components replaced.

For a entity pair (h,t), we consider its direct relation and multiple-step relation paths. The score function of PTransD is defined as

$$\begin{aligned} G(h,r,t) &= ||h_\perp + r - t_\perp||_{L_1/L_2} \\ &\quad + \frac{1}{Z} \sum_{p\in P(h,t)} R(p|h,t)||p - r||_{L_1/L_2} \end{aligned} \tag{15}$$

and the score function is further defined as

$$\begin{aligned} G(h,r,t) &= E(h,r,t) + E(h,P,t) \\ &= ||(r_p h_p^\top + I^{m*n})h + r - (r_p t_p^\top + I^{m*n})t||_{L_1/L_2} \\ &= \frac{1}{Z} \sum_{p\in P(h,t)} R(p|h,t)||p - r||_{L_1/L_2} \end{aligned} \tag{16}$$

where h_p, r_p, t_p are projection vectors which are used to construct dynamic mapping matrices. $A^\top$ is the transport matrix of A. We use I^{m*n} to denote the identity matrix of size m $\times$ n. $R(p|h,t)$ indicates the reliability of the relation path p given the entity pair (h,t), $\sum_{p\in P(h,t)} R(p|h,t)$ is a normalization factor.

2.3 Training Method and Implementation Details

Existing knowledge graphs only contain correct triples. We get the set of invalid triples by replacing one of three components of original valid triple (h,r,t). When corrupting the triple, we follow (Wang et al. 2014) [13] and assign different probabilities for head/tail entity replacement. For those 1-to-N, N-to-1 and

N-to-N relations, by giving more chance to replace "one" side, the chance of generating false-negative instances will be reduced. In experiments, we initialize entity vectors and relation vectors by the training result of TransE [3] to avoid overfitting.

For optimization, we use stochastic gradient descent (SGD) to minimize the loss function. We also enforce constraints on the norms of the embeddings h, r, t. That is, we set $||h||_2 \leqslant 1, ||t||_2 \leqslant 1, ||r||_2 \leqslant 1, ||h_p||_2 \leqslant 1, ||r_p||_2 \leqslant 1, ||t_p||_2 \leqslant 1, ||M_{rh}h||_2 \leqslant 1, ||r_\perp||_2 \leqslant 1, ||M_{rt}t||_2 \leqslant 1$. The head entity vector and the tail entity vector are projected from the entity semantic space R^n to the relation semantic space R^m through the mapping matrices M_{rh} and M_{rt}. It obvious that projected vectors rely on entities and relations.

In addition, there are some implementation details that will influence the performance of representation learning. In this paper, we add reverse relations for each relation in knowledge graph and We just consider 2-step relation paths existed in knowledge graph.

3 Experiments and Analysis

3.1 Data Set

Freebase is a huge and growing knowledge graph of a large number of the world facts, there are currently around 1, 2 billion triples and more than 80 million entities. In this paper, we adopt the subset of Freebase: FB15K (Bordes et al. 2014). Table 1 lists statistics and triple types of the FB15K dataset.

Table 1. Statistics of dataset FB15K.

Dataset	Ent	Rel	Train	Test	Valid	1-to-1	1-to-N	N-to-1	N-to-N
FB15K	14,951	1,345	483,142	59,071	50,000	26.2 (%)	22.7 (%)	29.3 (%)	22.8 (%)

In experiments, we use approach in TransH (Wang et al. 2014) [13] to replace the head and tail entity when corrupting the triple, which depends on the mapping property of the relations.

For comparison, we select all methods in (Lin et al., 2015) [1] as our baselines and use their reported results directly since the evaluation dataset is identical. The learning process of PTransD is carried out using stochastic gradient descent (SGD). To avoid overfitting, we initialize entity and relation embeddings with results of TransE [3], and initialize relation matrices as identity matrices.

3.2 Link Prediction

Link prediction [8] is to predict the missing h or t for a golden triple (h, r, t). In this task, we remove the head entity or tail entity and then replace it with all the entities of the dictionary in turn for each triple in test set. We first compute

Table 2. Evaluation results on entity prediction.

Metric	Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filter
RESCAL [15]	828	683	28.4	44.1
SE [6]	273	162	28.8	39.8
SME(linear) [2]	274	154	30.7	40.8
SME(bilinear) [2]	284	158	31.3	41.3
LFM [16]	283	164	26.0	33.1
TransE [3]	243	125	34.9	47.1
TransH [13]	212	87	45.7	64.4
TransR [7]	198	77	48.2	68.7
TransE(our)	235	136	51.6	71.4
TransD [9]	221	78	51.9	77.4
PTransE(ADD,2-step) [1]	200	54	51.8	83.4
PTransE(MUL,2-step) [1]	216	67	47.4	77.7
PTransE(RNN,2-step) [1]	242	92	50.6	82.2
PTransE(ADD,3-step) [1]	207	58	51.4	84.6
PTransD(ADD,2-step)	**147.32**	**21.04**	**54.97**	**92.58**

the scores of corrupted triples and then rank them in descending order, the rank of correct entity is finally store.

For evaluation, we use two ranking measures: the mean of those predicted ranks and the Hits@10, i.e. the proportion of correct ranked in the top 10. We call the original setting "Raw". Consider that a corrupted triple may also exist in knowledge graphs, the corrupted triple should be regarded as a correct triple. To avoid such a misleading behavior, we propose to remove from the list of corrupted triples all the triples that appear either in the training, validation or test set. We call this evaluation setting"Filter". In this paper, we provide mean rank and Hits@10 of the two settings.

For experiments of PTransD, we select learning rate λ among $\{0.1, 0.01, 0.001\}$, the dimensions of entity embedding m and relation embedding n among $\{20, 50, 100\}$, and the margin γ among $\{1, 2, 4\}$. The best configuration is determined according to the mean rank in validation set. The optimal configurations are $\lambda = 0.001$, the margin $\gamma = 1, m = n = 50$ and taking L_1 as dissimilarity. For the FB15K dataset, we traverse to training for 1000 rounds.

Evaluation results of entity prediction are show in Table 2. For that the addition operation and 2-step paths outperforms other composition operations in both mean rank and Hits@10, we just consider 2-step relation paths and use addition operation to represent the relation path.

From Table 2 we observe that: (1) PTransE [1] gets surprisingly low mean rank. It indicates that PTransD handles complicated internal correlations of entities and relations in knowledge graphs better than other methods. (2) In Hits@10, PTransD significantly and consistently outperforms other baselines including PTransE [1] and TransD [9].

As defined in (Bordes et al. 2013), relations in knowledge graphs can be divided into various types according to their mapping properties such as 1-to-1, 1-to-N, N-to-1, N-to-N. In order to further observe the performance of PTransD and other methods facing different complex relations, Table 4 shows that detailed results by mapping properties of relations on FB15K.

Table 3. Evaluation results by mapping properties of relations (%)

Tasks	Prediction head entities (Hits@10)				Prediction tail entities (Hits@10)			
Relation category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
SE [6]	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME(linear) [2]	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME(bilinear) [2]	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE [3]	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH [13]	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR [7]	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
TransD [9]	86.1	95.5	39.8	78.5	85.4	50.6	**94.4**	81.2
PTransE(ADD,2-step) [1]	91.0	92.8	60.9	83.8	91.2	74.0	88.9	86.4
PTransE(MUL,2-step) [1]	89.0	86.8	57.6	79.8	87.8	71.4	72.2	80.4
PTransE(RNN,2-step) [1]	88.9	84.0	56.3	84.5	88.8	68.4	81.5	86.7
PTransE(ADD,3-step) [1]	90.1	92.0	58.7	86.1	90.7	70.7	87.5	88.7
PTransD(ADD,2-step)	**91.79**	**96.51**	**82.48**	**92.47**	**91.79**	**94.88**	89.58	**94.31**

From Table 3 we can conclude that, where the head entities are predicted: (1) PTransD significantly outperforms other baselines including TransD [9] and PTransE [1] when dealing with the 1-to-1, N-to-1 and N-to-N relations types. (2) The performance of PTransD when dealing with the 1-to-N relations type outperform than other baselines. When the tail entities are predicted: (1) Compared with other models, PTransD is a more fine-grained model which considers the 1-to-1 and 1-to-N types of relations. (2) PTransD has poor ability of knowledge representation when dealing with N-to-1 type of relations, and it is inferior to TransD [9] model. (3) PTransD model performs well in dealing with N-to-N type of relations. It is observed that, on some mapping types of relations, PTransD achieves significant improvement as compared with other models. However, the performance of PTransD is not very well with respect to some types of relations [6].

3.3 Relation Prediction

Relation prediction aims to predict the relation of two entities, which is an important information source to enrich knowledge graphs. We use FB15K as our dataset. We use the score function to rank the candidate relations in relation prediction. Evaluation results are showed in Table 4, where we report Hits@1 instead of Hits@10 for comparison, because Hits@10 for both TransE [3] and PTranE [1] exceeds 95%. Similar to Hits@10, Hits@1 represents the proportion of correct ranked in the number one.

The best configurations of PTransD for relation prediction is $\lambda = 0.001$, $\gamma = 1$, $m = n = 50$ and take L_1 as dissimilarity.

Table 4. Evaluation results on relation prediction (%)

Metric	Mean Rank		Hits@1	
	Raw	Filter	Raw	Filter
TransE [3]	2.8	2.5	65.1	84.3
PTransE(ADD,2-step) [1]	1.7	1.2	69.5	93.5
PTransE(MUL,2-step) [1]	2.5	2.0	66.3	89.0
PTransE(RNN,2-step) [1]	1.9	1.4	68.3	93.2
PTransE(ADD,3-step) [1]	1.8	2.5	68.5	94.0
PTransD(ADD,2-step)	**1.5**	**1.1**	**70.83**	**94.76**

From Table 4, we can observe that: (1) PTransD (in mean rank respect) is lower than the results of TransE [3] and PTransE [1] models under both raw and filter settings. (2) In Hits@1, PTransD obtains 0.2 to 0.5% points higher than other models. That is, PTransD handles complicated internal correlations of entities and relations in knowledge graphs better than other methods. Obviously, PTransD significantly and consistently outperforms other baselines, which indicates that embedding by relation paths and dynamic mapping matrix provide a good supplement for representation learning.

4 Conclusions and Future Work

This paper proposes a noval representation learning method for knowledge graphs named PTransD, which embeds both entities and relations into a low-dimensional space for their completion [5]. In PTransD, to take advantage of relation paths, we consider multiple-step relation paths between entities and use the addition operation to represent relations paths for optimization; We divide entities and relations into different semantics spaces and construct mapping matrix dynamically. Experimental results on FB15K dataset show that PTransD outperforms TransE [3], TransD [9] and PTransE [1] on two tasks including link prediction and relation prediction.

As shown in Tables 2 and 3, evaluation results on some types of relations (e.g. 1-to-N) preforms not very well. One possible reason is that the way to replace the head and tail entity provides negative effect to the evaluation results. In the future, we will explore a new approach to replace the head and tail entity.

Besides, we will extend PTransD to better deal with complicated scenarios of knowledge graphs.

Acknowledgments. This work was partially supported by National Key R&D Program of China (No. 2016YFD0101903), National Natural Science Foundation of China (No. 61702386, 61672398), Major Technical Innovation Program of Hubei Province (No. 2017AAA122), Key Natural Science Foundation of Hubei Province of China (No. 2017CFA012), Applied Fundamental Research of Wuhan (No. 20160101010004), Fundamental Research Funds for the Central Universities (WUT:2018IVB047) and Excellent Dissertation Cultivation Funds of Wuhan University of Technology (2017-YS-061).

References

1. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge graphs. In: The Conference on Empirical Methods in Natural Language Processing (EMNLP 2015) (2015)
2. Bordes, A., Glorot, X., Weston, J., et al.: Joint learning of words and meaning representations for open-text semantic parsing. In: AISTATS 2012, vol. 22, pp. 127–135 (2012)
3. Bordes, A., Usunier, N., Garcia-Duran, A., et al.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp. 2787–2795 (2013)
4. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
5. Xie, R., Liu, Z., Jia, J., et al.: Representation learning of knowledge graphs with entity descriptions. In: AAAI 2016, pp. 2659–2665 (2016)
6. Bordes, A., Weston, J., Collobert, R., et al.: Learning structured embeddings of knowledge graphs. In: Conference on Artificial Intelligence (2011). (EPFL-CONF-192344)
7. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: The 29th AAAI Conference on Artificial Intelligence (AAAI 2015) (2015)
8. Yang, B., Yih, W., He, X., et al.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
9. Ji, G., He, S., Xu, L., et al.: Knowledge graph embedding via dynamic mapping matrix. In: ACL, vol. 1, pp. 687–696 (2015)
10. Mikolov, T., Chen, K., Corrado, G., et al.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
11. Yang, C., Liu, Z., Zhao, D., et al.: Network representation learning with rich text information. In: IJCAI 2015, pp. 2111–2117 (2015)
12. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Lechevallier, Y., Saporta, G. (eds.) Proceedings of COMPSTAT'2010, pp. 177–186. Physica-Verlag, Heidelberg (2010). https://doi.org/10.1007/978-3-7908-2604-3_16
13. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of AAAI, pp. 1112–1119 (2014)
14. Fan, M., Zhou, Q., Chang, E., et al.: Transition-based knowledge graph embedding with relational mapping properties. In: PACLIC 2014, pp. 328–337 (2014)
15. Egger, P., Pfaffermayr, M.: The proper panel econometric specification of the gravity equation: a three-way model with bilateral interaction effects. Empir. Econ. **28**(3), 571–580 (2003)

16. Jenatton, R., Roux, N.L., Bordes, A., et al.: A latent factor model for highly multi-relational data. In: Advances in Neural Information Processing Systems, pp. 3167–3175 (2012)
17. Duan, P., Wang, Y., Xiong, S., Mao, J.: Space projection and relation path based representation learning for construction of geography knowledge graph. In: China Conference on Knowledge Graph and Semantic Computing, CCKS 2016 (2016)
18. Bollacker, K., Evans, C., Paritosh, P., et al.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250. ACM (2008)
19. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
20. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706. ACM (2007)