Modeling IsA Relations via Box Structure for Knowledge Graph Embedding

Yao Dong[1,2], Lei Wang[1(✉)], Ji Xiang[1], and Kai Liu[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{dongyao,wanglei,xiangji,liukai}@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

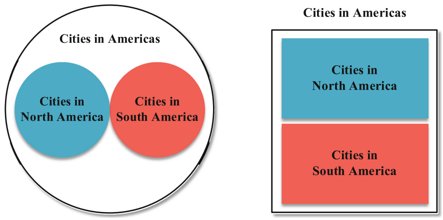
Abstract. Knowledge graph completion (KGC) aims to predict missing connections by mining information already present in a knowledge graph (KG). Predicting such connections is heavily dependent on the inference patterns. *IsA* relations (i.e., `instanceOf` and `subclassOf`) play an essential part in inferencing the composition pattern. Some existing methods already exploit *isA* relations. However, most of them learn insufficient representations, which may limit the performance. To address this issue, we propose a box-based knowledge graph embedding model called **IBKE**, in which concepts are embedded as boxes, and instances are represented by vectors in the same semantic space. According to the relative positions of elements, IBKE can naturally formulate *isA* relations. In addition, we introduce a random update strategy (RUS) for optimizing training, which updates embeddings in a probability pattern. Experimental results on benchmark datasets show that IBKE outperforms most existing state-of-the-art methods, and demonstrate the effectiveness of RUS.

Keywords: Knowledge graph embedding · Link prediction · Box

1 Introduction

Knowledge graphs (KGs) are structured facts of the real world, where nodes represents entities and edges between nodes represents relations. Large-scale KGs such as WordNet [14], YAGO [20] and Freebase [3] find applications in a variety of downstream tasks including machine translation [31], relation extraction [24], question answering [9] and recommender systems [29]. Although KGs may contain millions of triples, most existing KGs are incomplete. Therefore, much research work has been devoted to link prediction task, which is also known as knowledge graph completion (KGC). The target of link prediction is to predict missing facts in KG based on the existing links. An effective solution for KGC is knowledge graph embedding (KGE), which learns embeddings in a continuous low-dimensional vector space, and predicts missing links by evaluating the similarity of facts.

J. Gama et al. (Eds.): PAKDD 2022, LNAI 13281, pp. 303–315, 2022.
https://doi.org/10.1007/978-3-031-05936-0_24



(a) TransC (b) IBKE

Fig. 1. Space utilization.

State-of-the-art KGE models can be broadly categorised as translational models [4,10,12,26], semantic matching models [11,23,28] and deep learning models [6,16,25]. Most approaches focus on translational models in early times, which provide competitive performance with fewer parameters. Afterwards, several methods turn to semantic matching models, which achieve better performance by matching latent semantics of entities and relations. Recently, deep learning models for KGC have received increasing research attention. Such models generally achieve more outstanding performance on account of the larger parameters.

Despite achieving remarkable performance, most existing methods still regard both instances and concepts as entities to make a simplification, which leads to the following two drawbacks: **insufficient concept representation** and **lacking transitivity** of ***isA*** relations. To address these issues, TransC [13] is proposed as the first KGC model for differentiating concepts and instances, which encodes each concept as a hypersphere and each instance as a vector. Although modeling concepts via hyperspheres can building the transitivity of *isA* relations, it still result in the **insufficient concept representation**. A typical case is shown in Fig. 1(a). Commonly, parent class *Cities in Americas* can be exactly divided into two disjoint subclasses: *Cities in North America* and *Cities in South America*. However, TransC cannot take full advantage of space in parent class *Cities in Americas* under any circumstances, which means the representations of subclass concepts are insufficient. Furthermore, the blank space in the hypersphere of *Cities in Americas* lacks practical significance, which may lead to weak interpretability.

The problem of insufficient representation gives rise to the box structure [19]. Boxes can be regarded as the extended hyperspheres, which have different radii in each dimension. Similarly, boxes can easily deal with *isA* relations. Due to the flexibility of hyper-rectangles, boxes need only a slight effort to fill the gaps between the parent class and subclasses. Thus, boxes not only have more promising representation power but also reserve the superiority of hyperspheres.

In this paper, we propose a new method called **IBKE** for knowledge graph embedding. IBKE encodes each concept as a box (hyper-rectangle), while instances and relations are encoded as vectors. Further, we utilize relative positions between instances and concepts to model *isA* relations. Specifically, IBKE represents `instanceOf` relation by checking whether an instance vector is inside the box. For `subclassOf` relation, we enumerate four relative positions and define different score functions for three non-target cases: **disjoint**, **intersect** and **inverse**. Moreover, we introduce a new parameter update method called Random update strategy (RUS) for optimizing, which randomly updates embeddings according to two update thresholds. Note that RUS has good generalization ability for closed-region models.

In summary, **our contributions** are listed as follows:

- We propose IBKE, to the best of our knowledge, the first method using box structure to distinguish instances and concepts for modeling *isA* relations.
- We present a random update strategy, which enhances the representation power by updating parameters in a probability pattern.
- Through extensive experiments on two datasets, we show that IBKE achieves state-of-the-art performance in most cases. Besides, we analyze the random update strategy in detail and prove its effectiveness.

2 Related Work

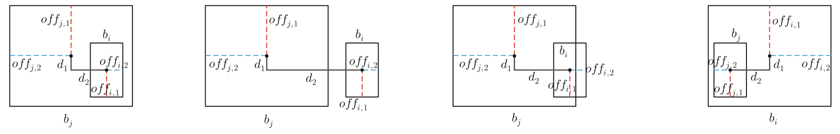
In this section, we give an overview of KGE models for link prediction, and divide previous methods into four categories.

Translational Models. TransE [4] is the first translational model, which encodes entities and relations as vectors based on the principle $\mathbf{h} + \mathbf{r} = \mathbf{t}$, where $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$ denotes head entity, relation and tail entity, respectively. Then, several variants are proposed to solve the drawbacks of TransE, including TransH [26], TransR [12] and TransD [10]. By introducing manifold-wise modeling, ManifoldE [27] remedys the N-N problem in TransE. TorusE [7] expands the embedding space to a Non-Euclidean space, i.e., torus. RotatE [21] first regards translations as rotations from head entity to tail entity in complex plane.

Semantic Matching Models. RESCAL [18] is the first bilinear model that can perform collective learning, which is prone to overfitting. Hence, DistMult [28] simplifies RESCAL by using a diagonal matrix. ComplEx [23] extends DistMult to the complex domain for modeling antisymmetric relations. HolE [17] combines the quintessence in DistMult and ComplEx. Recently, SimplE [11] presents a simple enhancement of Canonical Polyadic (CP) decomposition, and TuckER [2] is based on Tucker decomposition. QuatE [30] first models relations as rotations in quaternion space to enable rich and expressive semantic matching.

Deep Learning Models. ConvE [6], ConvKB [16] and InteractE [25] use convolutional neural network to capture the interactions between entities and relations. In addition, KBGAT [15] learns graph attention-based embeddings by a *generalized* graph attention model.

Region-Based Models. Generally, region-based models encode elements by explicitly defining the regions. These elements can be both entities and relations.



(a) $d_n \leq off_{j,n}$-$off_{i,n}$ (b) $d_n \geq off_{i,n}+off_{j,n}$ (c) -$off_{i,n} \leq d_n$-$off_{j,n} < off_{i,n}$ (d) $off_{i,n} > off_{j,n}$

Fig. 2. Four relative positions between box b_i and b_j.

Using a hypersphere to encode each concept, TransC [13] first differentiates concepts and instances. BoxE [1] that provides a solution to multi-arity KGC, encodes each relation as a box, while encodes each entity as a point and the corresponding *translational bump*.

Our proposed model IBKE belongs to the translational models. IBKE shares similarities with TransC, in which both models can deal with *isA* relations by differentiating concepts and instances. However, there are two major differences between TransC and IBKE:

- **Modeling.** IBKE encodes each concept as a box instead of a hypersphere, which is used in TransC.
- **Training.** Compared to TransC, we propose the random update strategy, which randomly learns parameters.

Note that we provide a comprehensive analysis about the computational complexity of several representative models in the supplemental material.[1]

3 Methodology

In this section, we propose a novel embedding method IBKE and present a new algorithm random update strategy (RUS).

3.1 IBKE

Formally, a knowledge graph is denoted by $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{S}\}$. Entity set $\mathcal{E}$ consists of instance set $\mathcal{I}$ and concept set $\mathcal{C}$, i.e., $\mathcal{E} = \mathcal{I} \cup \mathcal{C}$. Relation set $\mathcal{R} = \{r_i, r_c\} \cup \mathcal{R}_r$, where r_i represents `instanceOf` relation, r_c represents `subclassOf` relation and $\mathcal{R}_r$ denotes the set of other relations. Therefore, the triple set $\mathcal{S}$ can be divided into three disjoint subsets according to the relation type: relational triple set $\mathcal{S}_r$, `instanceOf` triple set $\mathcal{S}_i$ and `subclassOf` triple set $\mathcal{S}_c$.

Given a knowledge graph $\mathcal{G}$, KGC aims at predicting the missing links in $\mathcal{G}$ by learning embeddings for instances, concepts and relations in the same vector space $\mathbb{R}^k$, where k denotes the dimension of vector space. In IBKE, for each instance $i \in \mathcal{I}$ and relation $r \in \mathcal{R}_r$, we learn a k-dimensional vector $\mathbf{i} \in \mathbb{R}^k$ and

[1] The supplemental material of our paper is available online: https://github.com/JensenDong/IBKE.

$\mathbf{r} \in \mathbb{R}^k$, respectively. For each concept $c \in \mathcal{C}$, we learn a box $b(\mathbf{cen}, \mathbf{off})$ with $\mathbf{cen}, \mathbf{off} \in \mathbb{R}^k$ denoting the box center and offsets of all dimensions, respectively.

Box structure is more flexible, but it also brings the challenge that it is difficult to measure nested boxes. Thus, we define different dimensional-wise score functions for `instanceOf`, `subclassOf` and relational triples.

Relational Triples. A relational triple denoted as (h, r, t) consists of one relation and two instances. IBKE learns k-dimensional vectors for instances and relations. Hence, we define the score function just like TransE as follows:

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2. \tag{1}$$

InstanceOf Triples. For an `instanceOf` triple (i, r_i, c), when it holds, the instance i should be inside the **box** b. However, there is another relative position which i is out of the **box** b. Therefore, we define the following score function for optimizing:

$$f_i(i, c) = \sum_{n=1}^{k} (|i_n - cen_n| - off_n), \tag{2}$$

where i_n, cen_n and off_n represent the n-th element of $\mathbf{i}$, $\mathbf{cen}$ and $\mathbf{off}$, respectively.

SubclassOf Triples. For a `subclassOf` triple (c_i, r_c, c_j), when it holds, the **box** b_i should be inside the **box** b_j (as shown in Fig. 2(a)). However, there are three other relative positions between **box** b_i and b_j, i.e., **disjoint**, **intersect**, and **inverse**. Distance between the centers of b_i and b_j in n-th dimension is defined as follows:

$$d_n = |cen_{i,n} - cen_{j,n}|, \tag{3}$$

where $cen_{i,n}$ and $cen_{j,n}$ denote the n-th dimension of cen_i and cen_j, respectively. Further, we define a specific score function for each condition.

- **Disjoint.** b_i is disjoint from b_j (as shown in Fig. 2(b)). The two boxes should be closer in optimization. Therefore, the score function is defined as follows:

$$f_c(c_i, c_j) = \sum_{n=1}^{k} (d_n + off_{i,n} - off_{j,n}), \tag{4}$$

  where $off_{i,n}$ and $off_{j,n}$ denote the n-th dimension of $\mathbf{off_i}$ and $\mathbf{off_j}$, respectively.
- **Intersect.** b_i intersects with b_j (as shown in Fig. 2(c)). Similarly, we define the score fuction like the first condition as follows:

$$f_c(c_i, c_j) = \sum_{n=1}^{k} (d_n + off_{i,n} - off_{j,n}). \tag{5}$$

– **Inverse.** b_i is inside b_j (as shown in Fig. 2(d)). This condition is exactly the opposite of our optimization objective, so we define the following score function to reduce $off_{j,n}$ and increase $off_{i,n}$:

$$f_c(c_i, c_j) = \sum_{n=1}^{k} (off_{i,n} - off_{j,n}). \tag{6}$$

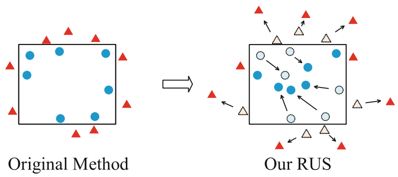


Fig. 3. Traditional method VS. RUS. Best view in colors. Red triangles represent negative instances and blue circles represent positive instances. Dotted triangles and circles represent their original positions. (Color figure online)

In experiments, we enforce constraints on embeddings, i.e., $\|\mathbf{h}\|_2 \leq 1$, $\|\mathbf{r}\|_2 \leq 1$, $\|\mathbf{t}\|_2 \leq 1$, $\|\mathbf{i}\|_2 \leq 1$, $\|\mathbf{cen}\|_2 \leq 1$ and $\forall n \in \{1, \ldots, k\}$, $off_n \leq 1$.

Optimization. We define a margin-based ranking loss function for relational triples as follows:

$$\mathcal{L}_r = \sum_{\xi \in \mathcal{S}_r} \sum_{\xi' \in \mathcal{S}'_r} [\gamma_r + f_r(\xi) - f_r(\xi')]_+, \tag{7}$$

where$[x]_+ \triangleq \max(0, x)$, ξ denotes a positive triple, ξ' denotes a negative triple and γ_r is the margin between positive triples and negative triples. Similarly, we define the loss function for `instanceOf` triples and `subclassOf` triples as follows:

$$\mathcal{L}_i = \sum_{\xi \in \mathcal{S}_i} \sum_{\xi' \in \mathcal{S}'_i} [\gamma_i + f_i(\xi) - f_i(\xi')]_+, \tag{8}$$

$$\mathcal{L}_c = \sum_{\xi \in \mathcal{S}_c} \sum_{\xi' \in \mathcal{S}'_c} [\gamma_c + f_c(\xi) - f_c(\xi')]_+. \tag{9}$$

We adopt stochastic gradient descent (SGD) to minimize the above loss functions, and use random update strategy (RUS) to randomly update embeddings.

Negative Sampling. Following Lv et al. [13], we randomly replace h or t to construct a negative triple (h', r, t) or (h, r, t'). (See details in supplemental material)

3.2 Random Update Strategy

During training `instanceOf` triples, the traditional method will stop updating parameters when score function $f_i(\xi) < 0$ for positive triples or $f_i(\xi') > 0$ for negative triples, which means that positive instances are inside the boxes or negative instances are outside the boxes, respectively. While the positive instances and negative instances are separated, they still gather around near the boundary of the boxes, i.e., the surface of hyper-rectangles. According to empirical regularity, instances and concepts should randomly distribute in the embedding space. Intuitively, we present a **R**andom **U**pdate **S**trategy (RUS) in place of the traditional algorithm. The comparison of these two algorithms is shown in Fig. 3. Note that we demonstrate this case in 2D for convenience.

In RUS, both positive triples and negative triples randomly update parameters depending on the score function f_i. We set two update thresholds $\phi_{pos}, \phi_{neg} \in [0, 1]$, where ϕ_{pos}, ϕ_{neg} denote positive update threshold and negative update threshold, respectively. For each positive training triple ξ, RUS updates parameters when $f_i > 0$ and updates parameters with probability ϕ_{pos} when $f_i \leq 0$. Similarly, for each negative training triple ξ', RUS updates parameters when $f_i < 0$ and updates parameters with probability ϕ_{neg} when $f_i \geq 0$. Details of RUS are summarized in supplemental material.

RUS is also applied to `subclassOf` triples. This strategy assists our model to separate positive and negative triples. Moreover, RUS can be generalized to a method that uses a structure of closed region such as hypersphere and box.

4 Theoretical Analyses

In this section, we provide some theoretical analyses of IBKE and Box structure. Note that all proofs for theorems can be found in the **supplemental material**.

4.1 Representation Power

Definition 1. *(**Filling Mode**) A filling mode is the way to stuff a box (hypersphere) with smaller ones. We define three types of filling modes.*

- ***Align** mode aligns the centers of the smaller boxes (hyperspheres) along each axis.*
- ***Compact** mode aligns the centers of the interlaced boxes (hyperspheres) to get a more compact spatial distribution.*
- ***Hybrid** mode is the mixture of **Align** mode and **Compact** mode.*

Theorem 1. *A hypersphere cannot be filled with several identical smaller hyperspheres without a single gap by any filling mode.*

Theorem 2. *A box can be filled with several identical smaller boxes without a single gap by a certain filling mode.*

Definition 2. *(**Representation Power**) The representation power of box (hypersphere) structure is the space utilization of embedding space.*

Theorem 3. *The representation power of box is superior to hypersphere.*

4.2 Inference Patterns

Knowledge graphs mainly consist of three relation patterns. We give their formal definitions here:

Definition 3. *A relation r is **symmetric(antisymmetric)** if*

$$\forall x, y \in \mathcal{E}, r(x,y) \Rightarrow r(y,x) \; (\, r(x,y) \Rightarrow \neg r(y,x)\,)$$

*A relation with such form is a **symmetry(antisymmetry)** pattern.*

Definition 4. *Relation r_1 is **inverse** to r_2 if*

$$\forall x, y \in \mathcal{E}, r_2(x,y) \Rightarrow r_1(y,x)$$

*Relations with such form is an **inversion** pattern.*

Definition 5. *Relation r_1 is **composed** of relation r_2 and relation r_3 if*

$$\forall x, y, z \in \mathcal{E}, r_2(x,y) \wedge r_3(y,z) \Rightarrow r_1(x,z)$$

*Relations with such form is a **composition** pattern.*

According to the above definitions, we provide a comprehensive analysis on IBKE in supplemental material and come to the following theorem:

Theorem 4. *IBKE can infer the antisymmetric, inversion and composition patterns.*

5 Experiments

In this section, we evaluate IBKE and RUS on link prediction [4]. In addition, we conduct a series of ablation experiments for RUS.

5.1 Experimental Setup

Datasets. Most previous models are evaluated on FB15k [4] and WN18 [4]. To address the test leakage problem in FB15k and WN18, FB15k-237 [22] and WN18RR [6] are constructed, which are subsets of FB15k and WN18, respectively. However, FB15k and FB15k-237 mainly consist of instances; WN18 and WN18RR mainly consist of concepts. The imbalance in the number of instances and concepts makes these four datasets inappropriate for testing the ability of distinguishing instances and concepts. Besides, *isA* relations are not explicitly given on these datasets. Even YAGO26K-906 [8] and DB111K-174 [8], which have explicitly given the *isA* relations, are not applicable. Both YAGO26K-906 and DB111K-174 suffer from the severe imbalance of instances and concepts, either. Moreover, these two datasets have test leakage problem and contain a large number of repeating triples. Hence in experiments, following TransC, we evaluate IBKE on benchmark dataset YAGO39K [13], which is constructed from another popular knowledge graph YAGO [20], and contains a number of instances and concepts. The statistics of YAGO39K are listed in supplemental material. In addition, we also evaluate IBKE on Countries dataset [5,21] to explicitly test the ability of inferring the composition pattern. It consists of three sub-tasks which increase in difficulty in a step-wise fashion. For more details about Countries, please see supplemental material.

Evaluation Protocol. Following Bordes et al. [4], the link prediction performance is reported on the standard evaluation metrics: Mean Reciprocal Rank (MRR) and Hits@N for $N = 1, 3, 10$. MRR is the mean reciprocal rank of correct triples. Hits@N is the proportion of correct triples whose rank is not larger than N. Note that an excellent embedding model should achieve a higher MRR and a higher Hits@N. We report the filtered results to avoid possibly flawed evaluation.

Table 1. Link prediction results on YAGO39K with $k = 100$ and $k = 200$. Best results are in **bold** and second best results are <u>underlined</u>.

Model	$k = 100$				$k = 200$			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE [4]	.248	.123	.287	.511	–	–	–	–
TransH [26]	.215	.104	.240	.451	–	–	–	–
TransR [12]	.289	.158	.338	.567	–	–	–	–
TransD [10]	.176	.089	.190	.354	–	–	–	–
HolE [17]	.198	.110	.230	.384	–	–	–	–
DistMult [28]	.362	.221	.436	.660	–	–	–	–
ComplEx [23]	.362	.292	.407	.481	–	–	–	–
SimplE [11]	.392	.283	.456	.590	.465	.367	.523	.644
TorusE [7]	.351	.295	.388	.449	–	–	–	–
TuckER [2]	.270	.187	.290	.428	.427	.315	.477	.653
KBGAT [15]	.469	.351	.539	.692	.475	.357	.543	.699
QuatE [30]	.399	.273	.452	.659	–	–	–	–
RotatE [21]	.504	.413	.560	.668	.552	.458	.611	.721
BoxE [1]	**.546**	**.462**	.598	.697	.566	.475	.626	<u>.726</u>
TransC [13]	.437	.299	.521	.700	.520	.406	.597	.720
IBKE (ours)	.522	.404	<u>.605</u>	**.731**	<u>.578</u>	<u>.487</u>	<u>.640</u>	**.729**
TransC-RUS	.448	.311	.534	<u>.704</u>	.531	.423	.602	.721
IBKE-RUS (ours)	<u>.532</u>	<u>.418</u>	**.613**	**.731**	**.582**	**.497**	**.641**	.725

Implementation. We select learning rate λ for SGD among {0.1, 0.01, 0.001}, the dimensionality of embedding space k among {20, 50, 100}, the three margins

γ_r, γ_i and γ_c among $\{0.1, 0.3, 0.5, 1, 2\}$, the two update thresholds ϕ_{pos} and ϕ_{neg} among $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. The optimal configurations on YAGO39K and Countries are listed in the supplemental material. To maintain comparison fairness, we train each model for 1000 epoches.

5.2 Results and Analysis

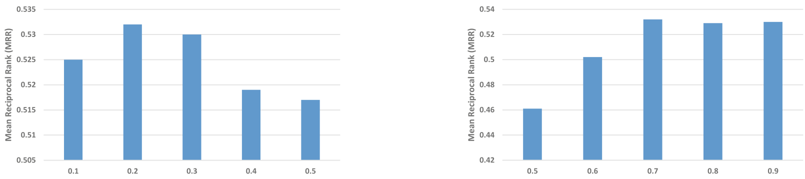
Evaluation results for relational triples are shown in Table 1. Note that we use publicly available source codes to reproduce results of comparison models, i.e., SimplE, TorusE, TuckER, KBGAT, QuatE, RotatE, BoxE and TransC. Other results are taken from [13]. From Table 1, we conclude that: (1) IBKE outperforms all baseline models in terms of Hits@3 and Hits@10. Results indicate that IBKE can get better performance by explicitly modeling *isA* relations. Distinguishing instances and concepts play a crucial part in learning embeddings. (2) The trend of performance with $k = 200$ is basically consistent with the performance with $k = 100$. We can see that IBKE outperforms all baseline models on all metrics when $k = 200$. The reason is that the representation power of box is more significant with a larger dimension. (3) The RUS works well for both IBKE and TransC, which implies a good scalability.

Comparison with TransC. IBKE achieves significant performance improvement. In specific, the improvement is 0.522–0.437 = 0.085 on MRR and +8.1% on Hits@1 over TransC when $k = 100$, which indicates that with a higher space utilization, the box structure is superior to hypersphere.

Comparison with BoxE. IBKE is only less competitive than BoxE in MRR and Hits@1 with $k = 100$, but outperforms BoxE on all metrics when $k = 200$. The reason is that IBKE encodes concepts as boxes and BoxE encodes relations as boxes. The number of concepts is larger than relations. Therefore, IBKE can capture more information with a larger dimension.

Random Update Strategy. To verify the effectiveness of RUS, we conduct a series of ablation experiments as shown in Table 1, Fig. 4 and Fig. 5(a). From Table 1, TransC-RUS achieves relative improvement of 0.448–0.437 = 0.011 on MRR and +1.7% on Hits@1 over TransC. Compared to IBKE, IBKE-RUS achieves relative improvement of 0.532–0.522 = 0.010 on MRR and +1.4% on Hits@1. Figure 4(a) shows that despite the epoch, models with RUS always outperform the corresponding ones without RUS. Moreover, as shown in Fig. 5, RUS can achieve better performance by using specific update thresholds.

Results on Countries S1/S2/S3. To further investigate the ability of inferring composition pattern, we evaluate our model on Countries dataset. In Table 2, we report the results with respect to the AUC-PR metric, which is commonly used in the literature. We can see that IBKE outperforms all the baseline models on S1 and S3, and obtains competitive performance on S2. Note that S3 is the most difficult task.

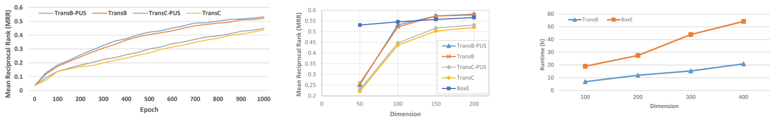


(a) Different ϕ_{pos} with $\phi_{neg} = 0.7$. (b) Different ϕ_{neg} with $\phi_{pos} = 0.2$.

Fig. 4. Performance with different update thresholds on MRR.

Table 2. Link prediction results of Countries datasets. Best results are in **bold**.

Model	Countries (AUC-PR)		
	S1	S2	S3
DistMult	**1.00**	0.72	0.52
ComplEx	0.97	0.57	0.43
ConvE	**1.00**	0.99	0.86
RotatE	**1.00**	**1.00**	0.95
IBKE	**1.00**	0.99	**0.96**



(a) (b) (c)

Fig. 5. (a) Learning curves of IBKE, IBKE-RUS, TransC, and TransC-RUS; (b) Performance versus dimensionality; (c) Runtime analysis.

Robustness Experiment. We evaluate the dependence of IBKE on dimensionality. Experimental results are shown in Fig. 4(b), in which we can conclude that: (1) Compared to IBKE, BoxE can obtain competitive performance with a smaller dimension. (2) When $k \geq 150$, IBKE can achieve state-of the-art performance relative to most models.

Running Time Analysis. We train IBKE-RUS on the CPU and BoxE on a single Tesla V100 GPU. Results are shown in Fig. 4(c), in which we can see that each runtime of IBKE is less than BoxE with same embedding dimension. Furthermore, as the dimension grows, so does the runtime gap between IBKE and BoxE. Hence, IBKE is more efficient than BoxE.

6 Conclusion

In this paper, we propose IBKE, which introduces a new use of box for knowledge graph completion. IBKE applies box structure to model concepts. Instances and relations are both embedded as vectors. We also propose a new parameters update method named random update strategy for randomly updating embeddings. Experimental results show that IBKE outperforms most state-of-the-art baselines and has obvious advantages when inferring the composition pattern. By ablation experiments, we further prove the effectiveness of RUS. In future work, we will explore how to combine box and rotation.

Acknowledgments. This work was supported by the National Key Research and Development Program of China.

References

1. Abboud, R., Ceylan, I., Lukasiewicz, T., Salvatori, T.: BoxE: a box embedding model for knowledge base completion. In: NeurIPS, vol. 33 (2020)
2. Balazevic, I., Allen, C., Hospedales, T.: TuckER: tensor factorization for knowledge graph completion. In: EMNLP, pp. 5185–5194 (2019)
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD, pp. 1247–1250 (2008)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NeurIPS, vol. 26, pp. 2787–2795 (2013)
5. Bouchard, G., Singh, S., Trouillon, T.: On approximate reasoning capabilities of low-rank vector spaces. In: AAAI Spring Symposium Series (2015)
6. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: AAAI, vol. 32 (2018)
7. Ebisu, T., Ichise, R.: TorusE: knowledge graph embedding on a lie group. In: AAAI, vol. 32 (2018)
8. Hao, J., Chen, M., Yu, W., Sun, Y., Wang, W.: Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In: SIGKDD, pp. 1709–1719 (2019)
9. Hao, Y., et al.: An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In: ACL, pp. 221–231 (2017)
10. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: ACL, pp. 687–696 (2015)
11. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: NeurIPS, pp. 4284–4295 (2018)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI, vol. 29 (2015)
13. Lv, X., Hou, L., Li, J., Liu, Z.: Differentiating concepts and instances for knowledge graph embedding. In: EMNLP, pp. 1971–1979 (2018)
14. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
15. Nathani, D., Chauhan, J., Sharma, C., Kaul, M.: Learning attention-based embeddings for relation prediction in knowledge graphs. In: ACL, pp. 4710–4723 (2019)
16. Nguyen, T.D., Nguyen, D.Q., Phung, D., et al.: A novel embedding model for knowledge base completion based on convolutional neural network. In: NAACL, pp. 327–333 (2018)

17. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: AAAI, vol. 30 (2016)
18. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: ICML, vol. 11, pp. 809–816 (2011)
19. Ren, H., Hu, W., Leskovec, J.: Query2box: reasoning over knowledge graphs in vector space using box embeddings. In: ICLR (2019)
20. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW, pp. 697–706 (2007)
21. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. In: ICLR (2019)
22. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, pp. 57–66 (2015)
23. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML (2016)
24. Vashishth, S., Joshi, R., Prayaga, S.S., Bhattacharyya, C., Talukdar, P.: Reside: improving distantly-supervised neural relation extraction using side information. In: EMNLP, pp. 1257–1266 (2018)
25. Vashishth, S., Sanyal, S., Nitin, V., Agrawal, N., Talukdar, P.P.: Interacte: improving convolution-based knowledge graph embeddings by increasing feature interactions. In: AAAI, pp. 3009–3016 (2020)
26. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI, vol. 28 (2014)
27. Xiao, H., Huang, M., Zhu, X.: From one point to a manifold: knowledge graph embedding for precise link prediction. In: IJCAI, pp. 1315–1321 (2016)
28. Yang, B., tau Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases (2015)
29. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: SIGKDD, pp. 353–362 (2016)
30. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. In: NeurIPS, pp. 2735–2745 (2019)
31. Zhao, Y., Zhang, J., Zhou, Y., Zong, C.: Knowledge graphs enhanced neural machine translation. In: IJCAI, pp. 4039–4045 (2020)