Group-constrained Embedding of Multi-fold Relations in Knowledge Bases*

Yan Huang[1,2][0000−0002−7925−4841], Ke Xu[1], Xiaoyang Yu[1], Tongyang Wang[1,2], Xinfang Zhang[1], and Songfeng Lu[2,3]

[1] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[2] Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen 518063, China
[3] School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China.

Abstract. Representation learning of knowledge bases aims to embed both entities and relations into a continuous vector space. Most existing models such as TransE, DistMult, ANALOGY and ProjE consider only binary relations involved in knowledge bases, while multi-fold relations are converted to triplets and treated as instances of binary relations, resulting in a loss of structural information. M-TransH is a recently proposed direct modeling framework for multi-fold relations but ignores the relation-level information that certain facts belong to the same relation. This paper proposes a Group-constrained Embedding method which embeds entity nodes and fact nodes from entity space into relation space, restricting the embedded fact nodes related to the same relation to groups with Zero Constraint, Radius Constraint or Cosine Constraint. Using this method, a new model is provided, i.e. Gm-TransH. We evaluate our model on link prediction and instance classification tasks, experimental results show that Gm-TransH outperforms the previous multi-fold relation embedding methods significantly and achieves excellent performance.

Keywords: Knowledge base · Representation learning · Multi-fold relation.

1 Introduction

Representation learning [7] has been proposed as a new approach for knowledge base representation and inference. It embeds entities and relations of a knowledge base into continuous vector space and preserves the structural information of original relational data. The representation of entities and relations are obtained by minimizing a global loss function involving all entities and relations. Compared with the traditional logic-based inference approaches, representation learning

* This work is supported by the Science and Technology Program of Shenzhen of China under Grant Nos. JCYJ20180306124612893, JCYJ20170818160208570 and JCYJ20170307160458368.

shows strong feasibility and robustness in applications such as semantic search, question answering, drug discovery and disease diagnosis.

Despite the promising achievements, most existing representation learning techniques (such as TransE [1], DistMult [18], ANALOGY [9] and ProjE [12] consider only binary relations contained in knowledge bases, namely triplets each involving two entities and one relation. For example, "Donald J. Trump is the president of America" consists of two entities "Donald J. Trump", "America" and a binary relation "president_of_a_country". However, a large amount of the knowledge in our real life are instances with multi-fold (n-ary, $n \geq 2$) relations, involving three or even more entities in one instance (such as "Harry Potter is a British-American film series based on the Harry Potter novels by author J. K. Rowling"). A general approach for this problem is to convert each multi-fold relation into multiple triplets with binary relations and learn the embedding of each triplet using the existing Trans(E, H, R) methods. Thus, an instance with a N-ary relation is converted to $\binom{N}{2}$ triplets [17]. Although such a conversion is capable of capturing part of the structures of multi-fold relations [11], it leads to a heterogeneity of the predicates, unfavorable for embedding. Wen et al. [17] advocates an instance representation of multi-fold relations and proposes a direct modeling framework "m-TransH" for knowledge base embedding. However, m-TransH treats fact nodes the same as general entity nodes and ignores the relation-level information that certain facts belong to the same relation.

In this paper, we first present a Group-constrained Embedding method which embeds entity nodes and fact nodes from entity space into relation space, restricting the embedded fact nodes related to the same relation to groups with three different constraint strategies, i.e. zero constraint, radius constraint and cosine constraint.

Then, using the Group-constrained Embedding method, we propose a new model "Gm-TransH" for knowledge base embedding with multi-fold relations. In terms of the three different constraint strategies, we advocate three variation of Gm-TransH, i.e. Gm-TransH:zero, Gm-TransH:radius, Gm-TransH:cosine. We conduct extensive experiments on the link prediction and instance classification tasks based on benchmark datasets FB15K [1] and JF17K [17]. Comparing with baseline models including Trans(E, H, R) and m-TransH, experimental results show that Gm-TransH outperforms the previous multi-fold relation embedding methods significantly and achieves state-of-the-art performance.

The main contributions of our work are as follows:

(a) Present a Group-constrained Embedding framework for multi-fold relation embedding, which embeds both entities and fact nodes into low dimensional vector space, forcing the fact embedding to be close to their corresponding relation vectors.

(b) We introduce three different types of group-constraints: Zero Constraint, Radius Constraint and Cosine Constraint. Their merits and demerits are analyzed empirically.

(c) We incorporate TransH model and propose a new model Gm-TransH and three variants Gm-TransH:Zero, Gm-TransH:Radius and Gm-TransH:Cosine for multi-fold relation embedding. Experimental results on link prediction

and instance classification tasks have proven the effectiveness of the three model variants.

(d) Clean the redundant data and generate a new subset G_{fact} for the JF17K datasets.

2 Related Work

2.1 Binary Relation Embedding

Most of the models proposed for knowledge base embedding are based on binary relations, datasets are in triplet representation.

TransE [1] sets $(h + r)$ to be the nearest neighbor of t when (h, r, t) holds, far away otherwise. **TransH** [16] is developed to enable an entity to have distinct distributed representations when involved in different relations. **TransR** [8] models entities and relations in distinct spaces and performs translation in relation space.

Besides TransE, TransH and TransR, many embedding methods based on binary relations are proposed, such as **MultiKE** [19], **RotatE** [14] and other translation embedding methods(e.g. **PTransE** [7], **TranSparse** [6], **KG2E** [3]), tensor factorization methods (e.g. **LFM** [4], **HolE** [10]) and neural network methods (e.g. **ProjE** [12], **Conv2D** [2], **NKGE** [15], **CrossE** [20]) and so on.

2.2 Multi-fold Relation Embedding

For knowledge bases with multi-fold relations, S2C conversion and decomposition framework [17] are usually used. Then, multi-fold relations are converted to triplets and treated as binary relations.

Wen et al. [17] proposes m-TransH model with a direct modeling framework to learn the embeddings of the entities and the n-ary relations, which generalizes TransH directly to multi-fold relations. In m-TransH, the cost function f_r is defined by

$$f_r(t) = \left\| \sum_{\rho \in M(R_r)} a_r(\rho) \mathbb{P}_{n_r}(t(\rho)) + b_r \right\|_2^2, t \in N^{M(R_r)} \tag{1}$$

Where $M(R_r)$ specifies a set of entity roles involved in relation R_r, N denotes all entities in a KB, R_r on N with roles $M(R_r)$ is a subset of $N^{M(R_r)}$, t is an instance of R_r and $t(\rho)$ indicates entity of role ρ. $\mathbb{P}_{n_r}(z)$ is the function that maps a vector $z \in U$ to the projection of z on the hyperplane with normal vector n_r, namely,

$$\mathbb{P}_{n_r}(z) = z - n_r^\top z n_r \tag{2}$$

n_r and b_r are unit length orthogonal vectors in U, $a_r \in \mathbb{R}^{M(R_r)}$ is a weighting function that

$$\sum_{\rho \in M(R_r)} a_r(\rho) = 0 \tag{3}$$

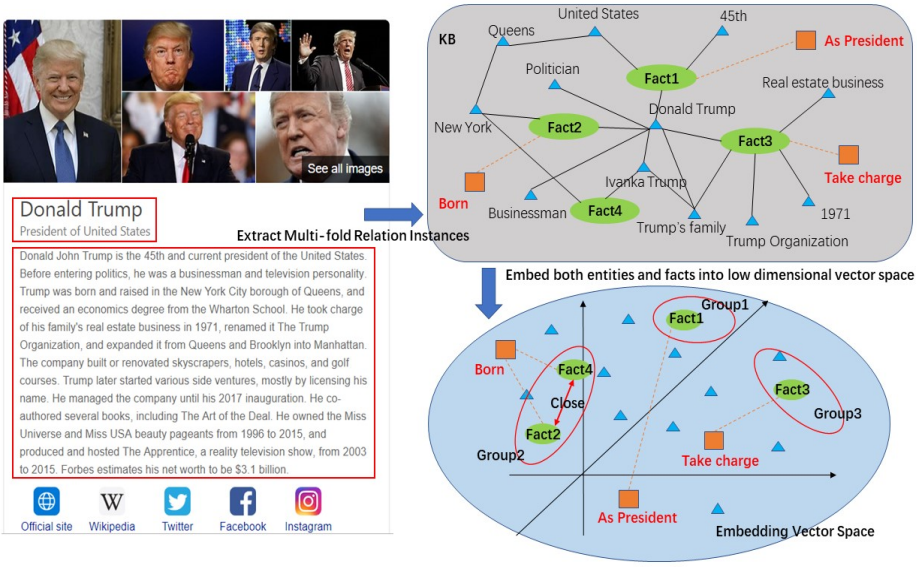

Fig. 1. Illustration of group-constrained embedding for multi-fold relations.

3 Group-constrained Embedding

3.1 Framework

Our framework for modeling multi-fold relations are shown in Figure 1. The knowledge extracted from raw text form instances of multi-fold relations in knowledge bases, we introduce fact node to represent each instance of particular relation and link the entities of the instance to corresponding fact node. These fact nodes may share some roles (i.e. entities) and relations. For example, in Figure 1, Fact2 and Fact4 share the same relation "born", i.e. both Donald Trump and Ivanka Trump were born in New York. We embed both entities and fact nodes into low dimensional vector space and let the embeddings of fact nodes of the same relation to be close, generating a group for each relation type, while groups of different relation to be far away from each other.

3.2 Optimizing Method

Converting multi-fold relations to binary relations results in a heterogeneity of the predicates, unfavorable for knowledge base embedding. M-TransH [17] treats fact nodes the same as general entity nodes and ignores the relation level information that certain facts belong to the same relation. Here, we propose an optimizing method called Group-constrained Embedding which embeds entity nodes and fact nodes from entity space into relation space, restricting the embedded fact nodes related to the same relation to a specific group. The cost function f_r is defined by $Eq.(4)$:

$$f_r(t) = \left\| \sum_{\rho \in M(R_r)} a_r(\rho) \mathbb{P}_{n_r}(t(\rho)) + b_r \right\|_2^2 + \beta * g_r(t), t \in N^{M(R_r)} \tag{4}$$

Where $g_r(t)$ is a penalty term used to restrict the embedded fact vectors and relation vectors. β is a decimal factor between 0 and 1 used to balance the

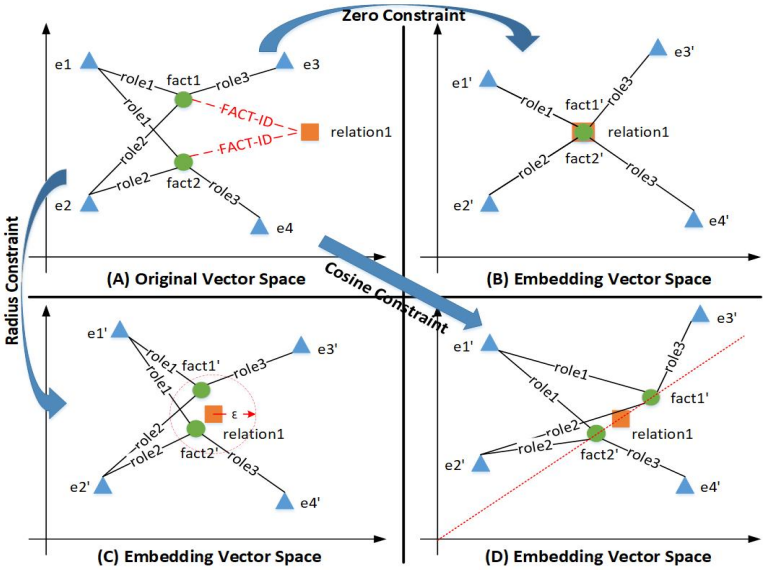

Fig. 2. Illustration of the three different strategies of group-constraint for multi-fold relation embedding in knowledge bases. We embed entities, facts and multi-fold relations from original vector space (i.e. graph A) to continuous vector space (i.e. graph B, C, D) using Zero Constraint, Radius Constraint or Cosine Constraint methods. Orange square indicates multi-fold relation, green circle indicates instance (i.e. fact node), blue triangle indicates general entities.

penalty and the loss. For simplicity, we use the offset vector b_r to represent the relation vector and measure the distance between fact embedding and relation vectors.

To solve the penalty term $g_r(t)$, we exploit three different types of constraints as below:

• **Zero Constraint**

Zero constraint adopts a rigorous constraint on the embedded fact vectors, and forces the Euclidean distance between the embedded fact vector $\mathbb{P}_{n_r}(e_{fact})$ and its corresponding relation vector b_r to be zero. Namely,

$$g_r(t) = \|b_r - \mathbb{P}_{n_r}(e_{fact})\|_2, t \in N^{M(R_r)} \tag{5}$$

This forces the fact embedding to be relation vector exactly, it can reduce the problem solving space and accelerate the model convergence. However, we argue that this rigorous constraint may reduce diversity and expressivity of the model.

• **Radius Constraint**

Radius constraint uses a trick to preserve model's diversity and expressivity, it adopts a relaxed constraint on the Euclidean distance between $\mathbb{P}_{n_r}(e_{fact})$ and b_r. If the fact is an positive instance of the relation r, we force the distance to be smaller than a very small positive number ϵ, otherwise much bigger than ϵ. In this way, we define $g_r(t)$ as $Eq.(6)$,

$$g_r(t) = \max(0, \|b_r - \mathbb{P}_{n_r}(e_{fact})\|_2 - \epsilon), t \in N^{M(R_r)} \tag{6}$$

• **Cosine Constraint**

Considering the drawback of Euclidean distance that each dimension contributes equally to the distance, we propose cosine constraint that exploits cosine distance as measurement and minimize the cosine distance of the embedded fact vector $\mathbb{P}_{n_r}(e_{fact})$ and its corresponding relation vector b_r. Namely,

$$g_r(t) = \cos\langle b_r, \mathbb{P}_{n_r}(e_{fact})\rangle, t \in N^{M(R_r)} \tag{7}$$

As depicted in Figure 2, we present an illustration of the three different types of group-constraints, which consists of 4 subgraphs, i.e. subgraph A, B, C and D. The first subgraph A shows the structure of the entities, facts and multi-fold relations in the original vector space. The other three subgraphs (i.e. subgraph B to D) show the Group-constrained Embedding of multi-fold relations with Zero Constraint, Radius Constraint and Cosine Constraint methods respectively.

In the original vector space in graph A, we have a 3-ary relation "relation1" (indicated by orange square) and two instances (indicated by green circle) with FACT-ID "$fact1$" and "$fact2$". Each of the two instances link with other three general entities (indicated by blue triangle) through different roles (i.e. $role1$, $role2$ and $role3$). We present 4 general entities $e1$, $e2$, $e3$ and $e4$ in graph A. We can see that $fact1$ and $fact2$ share the same entities on "$role1$" and "$role2$", differentiating on "$role3$".

In graph B, C, and D, we indicate the embedded vectors of instances and entities by adding a single quote to their names, e.g. the embedded vector of fact node "$fact1$" is marked as "$fact1'$". We indicate the embedded multi-fold relation "$relation1$" the same as it in the original vector space since they are the same vector and without a mapping operation.

Graph B shows the result of Group-constrained Embedding with Zero Constraint. As we force the Euclidean distance between the embedded fact vector "$fact1'$", "$fact2'$" and its corresponding relation vector "$relation1$" to be zero, these three vectors fall nearly into the same point in the embedded vector space. When using the radius constraint, as is shown in graph C, "$fact1'$" and "$fact2'$" fall into a hypersphere, "$relation1$" acts as the center of the sphere and the radius ϵ is a decimal number between 0 and 1. We can see that Radius Constraint degenerates to Zero Constraint when setting ϵ to 0. In graph C, we use the cosine distance as measurement, thus the angles of embedded vector "$fact1'$", "$fact2'$" and "$relation1$" are nearly the same, falling onto a straight line when projected to a hyperplane.

3.3 Proposed Model

Based on the Group-constrained Embedding method, we incorporate TransH model and propose a new multi-fold relation embedding model Gm-TransH as below, which consists of three variations corresponding to the three different types of constraints.

• **Group-constrained m-TransH (Gm-TransH)**

To solve the problem of m-TransH described above, we propose a new model that extends m-TransH to make the embedded fact vectors close to their corresponding relation vectors on the hyperplane.

In detail, we use the Radius Constraint for example, the embedded fact vectors that belong to the same relation lie in one hypersphere, the relation vector act

as the centre of the hypersphere, and the radius is a constant ϵ. Namely, if a fact is an instance of a relation, the distance between the embedded fact vector and the relation vector is forced to be smaller than ϵ on the hyperplane, otherwise much bigger than ϵ.

We call the above Group-constrained m-TransH model with Radius Constraint **Gm-TransH:radius**.

We can also use the Zero Constraint method and the Cosine Constraint method as substitute of the penalty term $g_r(t)$. Namely, with Zero Constraint method, the model Gm-TransH sets $g_r(t)$ to $Eq.(5)$ and denoted as **Gm-TransH:zero**.

Similarly, we use **Gm-TransH:cosine** to specify the Group-constrained m-TransH model with Cosine Constraint and set $g_r(t)$ to $Eq.(7)$.

3.4 Complexity Ayalysis

In Table 1, we compare the complexities of several models described in Related Work and the Gm-TransH models. For binary relation embedding models like SLM, NTN and Trans(E, H, R, D), we conduct a S2C conversion [17] for each instance with multi-fold relation, resulting in a collection of triplets with binary relations, which are appropriate for these models.

As listed in Table 1, the number of parameters of Gm-TransH models are same as m-TransH and lower than the binary relation embedding models. The time complexity (number of operations) of Gm-TransH models are higher than m-TransH and close to the TransH model.

As a matter of fact, the training time of the three different Gm-TransH:(radius, zero, cosine) models on the JF17K datasets with a dimension of 25 are about 45, 42 and 35 minutes respectively on a 32-core Intel Core i5-8300H 2.3GHz processor, which are close to transH and m-TransH(35 minutes) models.

4 Experiments and Analysis

4.1 Datasets

JF17K We use a cleaned and extended JF17K datasets [17] in our experiments. The original JF17K datasets were transformed from the full RDF formatted Freebase data. Denote the fact representation by F. Two datasets in instance representations for multi-fold relations, i.e. $T(F)$ (denoted by G), $T_{id}(F)$ (denoted by G_{id}) and a dataset in triplet representation for binary relations, i.e. $S2C(G)$ (denoted by G_{s2c}) were constructed, resulting in three consistent datasets, i.e. G, G_{id} and G_{s2c}.

However, as the provided JF17K datasets include many redundant samples, which may affect the results, we cleaned up the repetitive data at the beginning. In addition, the fact nodes (or CVT nodes) of a great quantity of instances were missing in the G_{id} dataset. We found the fact nodes indicated by role FACT-ID did not follow an 1-to-1 relationship to the multi-fold relations, which were not applicable for our proposed models. So we extended the G_{id} dataset and generated a fact node for each of these incomplete instances. Two instances which share same relation and entities except one role were assigned same fact

Table 1. Complexities (the number of parameters to train and the times of multiplication operations in each epoch) of several embedding models. N_e denotes the number of real entities, N_f denotes the number of fact nodes. N_r represents the number of multi-fold relations (i.e. $fold \geq 2$) and N_{r2} represents the number of binary relations. N_t represents the number of instances with multi-fold relations in the knowledge base. N_{t2} represents the number of triplets with binary relations. N_ρ denotes the sum of the folds of all instances with multi-fold relations. m and n are the dimensions of the entity and relation vector space respectively. d denotes the number of clusters of a relation. k is the number of hidden nodes of a neural network and s is the number of slice of a tensor.

Model	**# Parameters**	**# Operations**
SLM [13]	$O(N_e m + N_{r2}(2k + 2nk))$	$O((2mk + k)N_{t2})$
NTN [13]	$O(N_e m + N_{r2}(n^2 s + 2ns + 2s))$	$O(((m^2 + m)s + 2mk + k)N_{t2})$
TransE [1]	$O(N_e m + N_{r2} n)$	$O(N_{t2})$
TransH [16]	$O(N_e m + 2N_{r2} n)$	$O(2mN_{t2})$
TransR [8]	$O(N_e m + N_{r2}(m + 1)n)$	$O(2mnN_{t2})$
CTransR [8]	$O(N_e m + N_{r2}(m + d)n)$	$O(2mnN_{t2})$
TransD [5]	$O(2N_e m + 2N_{r2} n)$	$O(2nN_{t2})$
m-TransH [17]	$O((N_e + N_f)m + 2N_r n + N_\rho)$	$O(mN_\rho)$
Gm-TransH:zero	$O((N_e + N_f)m + 2N_r n + N_\rho)$	$O(m(N_\rho + N_t))$
Gm-TransH:radius	$O((N_e + N_f)m + 2N_r n + N_\rho)$	$O(m(N_\rho + N_t))$
Gm-TransH:cosine	$O((N_e + N_f)m + 2N_r n + N_\rho)$	$O(m(N_\rho + 3N_t))$

node. We call the extended set G_{fact} and divided it into training set $G^{\checkmark}_{fact}$ and testing set $G^{?}_{fact}$. The statistics of these datasets are shown in Table 2.

FB15K To verify the effectiveness of our models on a particular degenerated type of multi-fold (N-ary) relation, i.e. binary relation with $N = 2$, we also conduct instance classification task on FB15K dataset [1]. Since FB15K dataset is consist of triplets in binary relations only and has no information of fact nodes, we extend the FB15K dataset and attach an unique fact node to each triplet. Thus, we can use the extended FB15K dataset to train the proposed Gm-TransH model and test its performance while only binary relations holds. To compare with benchmark models for binary relations, we use the original FB15K dataset to train the NTN, TransE, TransH and TransR models. For convenience, we use "*Raw*" to denote the original FB15K dataset and use "*Ext*" to denote the extended FB15K dataset. Table 3 lists the statistics of the original and extended FB15K datasets.

4.2 Link Prediction

Link prediction aims to complete the missing entities for instances or triplets, i.e., predict one entity given other entities and the relation. For example, for triplet (h, r, t), predict t given (h, r) or predict h given (r, t). As for instances with multi-fold relations, the missing entity can be any one of the entities associated with the relation r. Link prediction ranks a set of candidate entities from the knowledge graph. We use the extended JF17K datasets in this task and compare with some of the canonical models including TransE, TransH, TransR and m-TransH.

Table 2. Statistics of the extended JF17K dataset.

Dataset	$\mathbf{G^{\checkmark}_{s2c}}/\mathbf{G^{?}_{s2c}}$	$\mathbf{G^{\checkmark}}/\mathbf{G^{?}}$	$\mathbf{G^{\checkmark}_{id}}/\mathbf{G^{?}_{id}}$	$\mathbf{G^{\checkmark}_{fact}}/\mathbf{G^{?}_{fact}}$
# Entities	17629/12282	17629/12282	17629/12282	17818/17818
# Relations	381/336	181/159	181/159	181/159
# Samples	118568/30912	89248/17842	93976/18318	36199/10560

Table 3. Statistics of the original and extended FB15K dataset.

Dataset	# Rel	# Ent	# Train	# Valid	# Test
FB15K(*Raw*)	1,345	14,951	483,142	50,000	59,071
FB15K(*Ext*)	1,345	19,966	483,142	50,000	59,071

Evaluation protocol In this task, for every instance in test set, we remove each of the entities and then replace it with the entities in the real entity (as opposed to fact entity) set in turn. For fairness, we replace only the real entities appeared in the instances and exclude the fact nodes. Dissimilarities of the corrupted instances are first computed via the proposed models and then sorted by ascending order. Then we use Hit@10(HIT) and Mean Rank (RANK) [1] ranked by the correct entities as the performance metrics to evaluate the proposed models. These two metrics are commonly used to evaluate the performance of knowledge base embeddings. Hit@10 computes the probability of the positive entities that rank up to the top 10% for all the entities. Mean Rank means the average position of the positive entities ranked.

Table 4. The models and datasets used for link prediction.

Experiment	**Model**	**Training Dataset**	**Testing Dataset**
TransE:triplet	TransE(bern)	$\mathbf{G^{\checkmark}_{s2c}}$	$\mathbf{G^{?}_{s2c}}$
TransH:triplet	TransH(bern)	$\mathbf{G^{\checkmark}_{s2c}}$	$\mathbf{G^{?}_{s2c}}$
TransR:triplet	TransR(bern)	$\mathbf{G^{\checkmark}_{s2c}}$	$\mathbf{G^{?}_{s2c}}$
m-TransH:inst	m-TransH	$\mathbf{G^{\checkmark}}$	$\mathbf{G^{?}}$
m-TransH:ID	m-TransH	$\mathbf{G^{\checkmark}_{id}}$	$\mathbf{G^{?}_{id}}$
Gm-TransH:zero	Gm-TransH	$\mathbf{G^{\checkmark}_{fact}}$	$\mathbf{G^{?}_{fact}}$
Gm-TransH:radius	Gm-TransH	$\mathbf{G^{\checkmark}_{fact}}$	$\mathbf{G^{?}_{fact}}$
Gm-TransH:cosine	Gm-TransH	$\mathbf{G^{\checkmark}_{fact}}$	$\mathbf{G^{?}_{fact}}$

Implementation We trained and tested eight kinds of models in this task, the training and testing datasets employed by each of the models as well as the model they train are shown in the Table 4.

Stochastic Gradient Descent is used for training, as is standard. We take $L2$ as dissimilarity and traverse all the training samples for 1000 rounds. Several choices of the dimension d of entities and relations are studied in our experiments: 25, 50, 100, 150, 200, 250. We select learning rate λ for SGD among 0.0015, 0.005, 0.01, 0.1, the balance factor β for Gm-TransH among 0.001, 0.01, 0.05 0.1, the

margin γ among 0.5, 1.0, 2.0, and the radius ϵ in Gm-TransH:radius among 0.01, 0.05, 0.1, 0.5, 1, 5, the batch size B among 120, 480, 960, 1920. The optimal configurations for the three different Gm-TransH models are Gm-TransH:zero: λ=0.0015, β=0.01, γ=0.5, d=150, B=960. Gm-TransH:radius: λ=0.0015, β=0.05, γ=1.0, ϵ=0.05, d=250, B=480. Gm-TransH:cosine: λ=0.0015, β=0.01, γ=1.0, d=200, B=1920.

Results Experimental results of link prediction on the cleaned and extended JF17K datasets are shown in Table 5, which shows the Hit@10 results and Mean Rank results of different embedding models with dimension 25, 50, 100, 150, 200, 250 respectively. The three Gm-TransH models outperform the Trans(E, H, R) models by a large margin on both Hit@10 and Mean Rank metrics. Compared to the m-TransH models, our models achieve an improvement on the probability of Hit@10 and get an approximate Mean Rank with m-TransH:inst. The results show that our approach is effective on improving the accuracy of link prediction via multi-fold relation embeddings. Furthermore, by contrast, Gm-TransH:zero outperforms Radius Constraint and Cosine Constraint on Hit@10 metric, showing that Zero Constraint is better for discrimination. Gm-TransH:cosine is best performed on Mean Rank metric and has higher overall optimizing ability.

Table 5. Experimental results (HIT/RANK) of link prediction on the extended JF17K dataset.

Experiment/DIM	**25**	**50**	**100**	**150**	**200**	**250**
TransE:triplet	29.43%/153.8	30.28%/159.5	31.05%/149.2	31.45%/145.6	29.53%/152.5	29.63%/155.5
TransH:triplet	35.42%/111.2	36.36%/111.7	35.51%/120.1	36.52%/109.2	35.29%/113.1	36.58%/123.5
TransR:triplet	35.35%/126.1	36.56%/104.9	36.51%/113.3	36.47%/106.7	35.56%/114.6	36.12%/126.9
m-TransH:inst	62.87%/**78.7**	66.54%/81.4	67.24%/**76.4**	68.16%/78.6	67.24%/82.2	67.51%/86.1
m-TransH:ID	73.37%/107.2	74.06%/109.9	77.51%/107.5	79.07%/106.1	78.55%/112.1	78.36%/105.3
Gm-TransH:zero	**76.73**%/80.1	**78.56**%/81.9	**82.17**%/78.5	**83.28**%/81.5	82.25%/79.8	**81.63**%/82.7
Gm-TransH:radius	75.52%/80.9	77.19%/80.2	81.05%/78.2	81.98%/78.3	**82.74**%/78.8	81.37%/80.7
Gm-TransH:cosine	74.29%/79.3	77.96%/**78.5**	80.01%/77.2	81.27%/**75.7**	81.30%/**78.5**	79.14%/**79.3**

Table 6. Evaluation accuracy(%) of instance classification.

Datasets	**FB15K**(*Raw*)	**FB15K**(*Ext*)	**FB17K**
NTN	68.2	—	51.3
TransE(unif/bern)	77.3/79.8	—	54.4/58.5
TransH(unif/bern)	74.2/79.9	—	55.6/59.1
TransR(unif/bern)	81.1/82.1	—	60.7/63.4
m-TransH:inst	—	83.2	72.5
m-TransH:ID	—	84.7	76.7
Gm-TransH:zero	—	90.4	88.2
Gm-TransH:radius	—	**92.3**	92.3
Gm-TransH:cosine	—	90.1	**92.6**

4.3 Instance Classification

Instance classification aims to judge whether a given instance is correct or not. This is a binary classification task, which has been explored in [13,16] for evaluation. In this task, we use the extended JF17K and FB15K datasets to

evaluate our models. For comparison, we select the NTN, TransE, TransH, TransR and m-TransH as baseline models.

Evaluation protocol For instance classification task, we follow the same protocol in NTN and TransH. Since the evaluation of classification needs negative labels, the JF17K and FB15K datasets both consist of positive instances only, we construct negative instances following the same procedure used for FB13 in [13]. For each golden instance, one negative instance is created.

We set a threshold δ_r for each relation r by maximizing the classification accuracies on the training set. For a given instance in the testing set, if the dissimilarity score is lower than δ_r, it will be classified as positive, otherwise negative.

Implementation For binary relation embeddings of triplets, we train and evaluate the NTN, Trans(E, H, R) models on the original FB15K dataset (denoted as Raw) and the G_{s2c} dataset of JF17K. We use the NTN code released by Socher [13] and the Trans(E, H, R) code released by [8] directly. For multi-fold relation embeddings of instances, we use the m-TransH code released by [17] and implement the Gm-TransH models to evaluate on extended FB15K(Ext) dataset and the G, G_{id}, G_{fact} datasets of JF17K respectively. We select the same hyperparameters as used in link prediction and get the average accuracy of 20 repeated trials.

Results Table 6 lists the evaluation results of instance classification in detail. We can observe that both on FB15K and JF17K datasets, the Gm-TransH models can reach to an accuracy of 90%, outperforming the baseline models including NTN, Trans(E, H, R) and m-TransH significantly. This shows our models can learn the relation-level information effectively and expressively. Moreover, from the results on the FB15K(Raw) and the FB15K(Ext) datasets, we see that even for binary relations, the Group-constrained Embedding models are practicable and reliable.

5 Conclusions and Future Work

We presented a group-constrained embedding framework with three different types of constraint strategies for multi-fold relations and proposed a new representation learning model, i.e. Gm-TransH. We evaluate the effectiveness and performance of the proposed models on extended FB15K and JF17K datasets. Experimental results show that the Gm-TransH models outperforms all baseline models on link prediction and instance classification task. In the future, we will explore more representation and embedding frameworks for the increasingly complicated data in knowledge bases, e.g. events and procedures, as well as incorporating the most recent advances in the learning of binary relations for multi-fold relation embedding.

References

1. Bordes, A., Usunier, N., Garciaduran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. neural information processing systems pp. 2787–2795 (2013)

2. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. national conference on artificial intelligence pp. 1811–1818 (2018)
3. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with gaussian embedding. conference on information and knowledge management pp. 623–632 (2015)
4. Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. neural information processing systems pp. 3167–3175 (2012)
5. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. international joint conference on natural language processing pp. 687–696 (2015)
6. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. national conference on artificial intelligence pp. 985–991 (2016)
7. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. empirical methods in natural language processing pp. 705–714 (2015)
8. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. national conference on artificial intelligence pp. 2181–2187 (2015)
9. Liu, H., Wu, Y., Yang, Y.: Analogical inference for multi-relational embeddings. international conference on machine learning pp. 2168–2178 (2017)
10. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. national conference on artificial intelligence pp. 1955–1961 (2016)
11. Rouces, J., De Melo, G., Hose, K.: Framebase: Representing n-ary relations using semantic frames. european semantic web conference pp. 505–521 (2015)
12. Shi, B., Weninger, T.: Proje: Embedding projection for knowledge graph completion. national conference on artificial intelligence pp. 1236–1242 (2017)
13. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. neural information processing systems pp. 926–934 (2013)
14. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. international conference on learning representations (2019)
15. Wang, K., Liu, Y., Xu, X., Lin, D.: Knowledge graph embedding with entity neighbors and deep memory network. international conference on learning representations (2019)
16. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. national conference on artificial intelligence pp. 1112–1119 (2014)
17. Wen, J., Li, J., Mao, Y., Chen, S., Zhang, R.: On the representation and embedding of knowledge bases beyond binary relations. international joint conference on artificial intelligence pp. 1300–1307 (2016)
18. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. international conference on learning representations (2014)
19. Zhang, Q., Sun, Z., Hu, W., Chen, M., Guo, L., Qu, Y.: Multi-view knowledge graph embedding for entity alignment. international joint conference on artificial intelligence (2019)
20. Zhang, W., Paudel, B., Zhang, W., Bernstein, A., Chen, H.: Interaction embeddings for prediction and explanation in knowledge graphs. web search and data mining pp. 96–104 (2019)