Learning Knowledge Graph Embeddings via Generalized Hyperplanes

Qiannan Zhu[1,2], Xiaofei Zhou*[1,2], JianLong Tan[1,2], Ping Liu[1,2], and Li Guo[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, China
[2] University of Chinese Academy of Sciences, School of Cyber Security, China
{zhuqiannan,zhouxiaofei}@iie.ac.cn

Abstract. For knowledge graph completion, translation-based methods such as Trans(E and H) are promising, which embed knowledge graphs into continuous vector spaces and construct translation operation between head and tail entities. However, TransE and TransH still have limitations in preserving mapping properties of complex relation facts for knowledge graphs. In this paper, we propose a novel translation-based method called translation on generalized hyperplanes (TransGH), which extends TransH by defining a generalized hyperplane for entities projection. TransGH projects head and tail embeddings from a triplet into a generalized relation-specific hyperplane determined by a set of basis vectors, and then fulfills translation operation on the hyperplane. Compared with TransH, TransGH can capture more fertile interactions between entities and relations, and simultaneously has strong expression in mapping properties for knowledge graphs. Experimental results on two tasks, link prediction and triplet classification, show that TransGH can significantly outperform the state-of-the-art embedding methods.

Keywords: Knowledge Representation,Knowledge Embedding,Knowledge Graph Completion.

1 Introduction

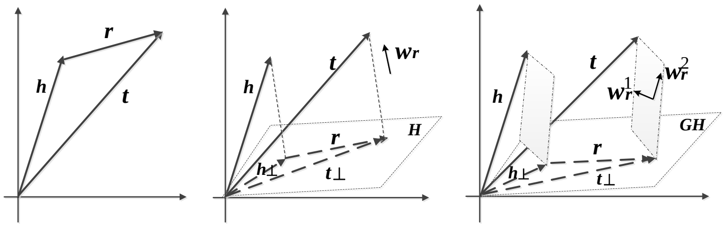
Knowledge graphs like Freebase [1], WordNet [15] and Google Knowledge Graph play extremely practical roles in numerous AI applications, such as Question Answering System [7] and Information Extraction [9]. A typical knowledge graph (KG) is a multi-relational directed graph, in which nodes represent entities and edges represent different types of relations. That is, a basic triplet fact (h, r, t) in KG represents that the relationship r links the head entity h and tail entity t. e.g., (Barack_Obama, Place_of_Birth, Hawai). Although there are huge amounts of structured data, a knowledge graph is factually far from completeness. Knowledge graph completion aims to predict new relational facts under supervision of the existing knowledge graph.

In the past decade, massive traditional approaches based on logic and symbol [16, 17] have been done for knowledge graph completion, but they are intractable and not enough convergence for large scale knowledge graphs. Recently an emerging approach

* Corresponding Author

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_48

called knowledge graph embedding, which embeds all objects(entities and relations) of a KG into a low-dimensional space, have highly attracted attention. Following this approach, many models described in Section ”Related Work” have been presented. Among these models, Trans(E, H, R and D) [5, 19, 12, 11] are fundamental and efficient while achieving state-of-the-art predictive performance. TransE [5] simply and directly build entity and relation embeddings by regarding a relation as translation from head entity to tail entity, but there are flaws in dealing with complex relations, such as reflexive, one-to-many, many-to-one, and many-to-many relations. To address these issues of TransE, TransH [19] considers some mapping properties of complex relations in embedding, and projects entity embeddings into relation-specific hyperplanes. But for TransH, there is only one normal vector used for modeling relation-specific hyperplane, which leads that entities and relations are still in the same space and a limit representation for mapping properties. TransR [12] regards to map entity embeddings into r-relation space with a transfer matrix, and TransD [11] uses the product of two projection vectors of an entity-relation pair to construct the transfer matrix. Such transfer matrix can build entity and relation embeddings in separate spaces and has more general representation for mapping properties, however, it will cost much more computations and memories on the mappings.



(a) TransE (b) TransH (c) TransGH

Fig. 1. Simple visualization of TransE, TransH and TransGH.

In this paper, we propose an expressive model named translation on generalized hyperplanes (TransGH) to promote TransH. Instead of the only one normal vector, TransGH uses a set of basis vectors to determine a generalized hyperplane. Figure 1 simply shows the differences of TransE, TransH and TransGH.

- TransE builds the translation from head embedding to tail embedding as $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when the triplet (h, r, t) holds.
- TransH projects entity embeddings into relation-specific hyperplanes characterized by one normal vector $\mathbf{w}_r$, and builds translation between the projected entities on the hyperplane as $\mathbf{h}_\perp + \mathbf{r} \approx \mathbf{t}_\perp$, where $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^T\mathbf{h}\mathbf{w}_r$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^T\mathbf{t}\mathbf{w}_r$.
- Different from TransH, TransGH uses a set of basis vectors $\{\mathbf{w}_r^1, \mathbf{w}_r^2, ..., \mathbf{w}_r^v\}$, $(v << |\mathbf{h}|)$ to determine a generalized relation-specific hyperplane, and the mappings of the entity embeddings on the hyperplane are $\mathbf{h}_\perp = \mathbf{h} - \sum_i {\mathbf{w}_r^i}^T\mathbf{h}\mathbf{w}_r^i$ and $\mathbf{t}_\perp = \mathbf{t} - \sum_i {\mathbf{w}_r^i}^T\mathbf{h}\mathbf{w}_r^i (i \in [1, v])$.

The basic idea of TransGH illustrated in Figure 1(c) is that for a given triplet (h, r, t), firstly the entity embeddings $\mathbf{h}$ and $\mathbf{t}$ are projected on the generalized hyperplane as $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$ with a set of basis vectors respectively, where the embedding $\mathbf{h}_\perp$ is expected to be close to the embedding $\mathbf{t}_\perp$ by adding the relation embedding $\mathbf{r}$.

Our contributions in this paper are: (1) We propose a novel model TransGH, which models each relation as a vector on the generalized hyperplane determined by a set of basis vectors. (2) TransGH has the similar parameters to TransH as it only extends one normal vector to a set of basis vectors, indicating that TranGH is applicable to large scale KGs. (3) In the two tasks of link prediction and triplet classification, TransGH has significant improvements comparing with previous Trans(E,H,R and D).

2 Related Work

2.1 Translation-based Models

Translation-based models usually embed entities and relations into a low-dimensional vector space, and enforce vector embeddings compatible under a score function $f(h, r, t)$. Different models have the different definitions of score functions. Below we briefly summarize some baseline translation-based models and give the corresponding score functions.

TransE [5] embeds entities and relations into the same space R^m, and interprets each relation as a translation vector from the head entity embedding to tail entity embedding. Hence the score function is defined as $f(h, r, t) =\parallel \mathbf{h} + \mathbf{r} - \mathbf{t} \parallel_2^2$ for a triplet (h, r, t). TransE is effective for one-to-one relations but has flaws in dealing with one-to-many, many-to-one and many-to-many relations.

To overcome the issues of TransE, TransH [19] projects entity embeddings into relation-specific hyperplanes to enable an entity has distinct representations when involved in different relations. It models each relation r as a vector $\mathbf{r}$ on the hyperplane with a normal vector $\mathbf{w}_r$, therefore the scoring function is defined as $f(h, r, t) =\parallel \mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp \parallel_2^2$. With $\parallel \mathbf{w}_r \parallel_2 = 1$, it is easily to get $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^T \mathbf{h} \mathbf{w}_r$, $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r{}^T \mathbf{t} \mathbf{w}_r$, and $\mathbf{h}, \mathbf{t}, \mathbf{r}, \mathbf{w}_r \in R^m$.

Both TransE and TransH embed entities and relations into the same vector space without considering that entities and relations are different types of objects. TransR/CtransR [12] regards entities and relations as completely different objects via embedding entities and relations into entity space R^m and relation spaces R^n, respectively. It maps entity embeddings from entity space to r-relation space with a mapping matrix $\mathbf{M}_r$. Then the score function is defined as $f(h, r, t) =\parallel \mathbf{h}_r + \mathbf{r} - \mathbf{t}_r \parallel_2^2$, where $\mathbf{h}_r = \mathbf{h}\mathbf{M}_r$, $\mathbf{t}_r = \mathbf{t}\mathbf{M}_r$ and $\mathbf{h}, \mathbf{t} \in R^m, \mathbf{r} \in R^n, \mathbf{M}_r \in R^{m \times n}$. CtransR is an extension of TransR, which divides all the entity pair(h, t) in the training data into multiple groups(clusters) and learns independent relation vector for each group.

TransD [11] is an improvement of TransR/CtransR, which considers the multiple types of entities and relations simultaneously. It replaces the transfer matrix by the product of two projection vectors of an entity-relation pair. Therefore score function is denoted as $f(h, r, t) =\parallel \mathbf{M}_{rh}\mathbf{h} + \mathbf{r} - \mathbf{M}_{rt}\mathbf{t} \parallel_2^2$, where $\mathbf{M}_{rh} = \mathbf{r}_p \mathbf{h}_p{}^T + \mathbf{I}^{n \times m}$, $\mathbf{M}_{rt} = \mathbf{r}_p \mathbf{t}_p{}^T + \mathbf{I}^{n \times m}$, and $\mathbf{h}, \mathbf{h}_p, \mathbf{t}, \mathbf{t}_p \in R^m, \mathbf{r}, \mathbf{r}_p \in R^n$.

Recently TransE-RS and TransH-RS [20] combine a limit-based scoring loss for learning knowledge embeddings, which have significant improvements compared to state-of-the-art baselines.

2.2 Other Models

Unstructured Molel(UM) [4] is a simplified version of TransE with considering the knowledge graph as none-relation and setting all relation vectors as $\mathbf{r} = 0$, which leads to the score function $f_r(h, r, t) =\| \mathbf{h} - \mathbf{t} \|$. Obviously, this model can not deal with the different relations.

Structured Embedding(SE) [6] interprets entities as vectors and each relation as two independent matrices $\mathbf{M}_r^h$ and $\mathbf{M}_r^t$ for projecting the head entity embedding and tail entity embedding. Then score function is $f_r(h, r, t) = -\|\mathbf{M}_r^h\mathbf{h} - \mathbf{M}_r^t\mathbf{t}\|$. SE can not capture the information between entities and relations since it uses the two separate matrices.

Latent Factor Model(LFM) [10, 18] encodes entities as vectors and sets each relation as a matrix. Each r-specific matrix is asymmetric and directly operates between two entity embeddings. The score function is $f(h, r, t) = \mathbf{h}^T\mathbf{M}_r\mathbf{t}$.

Semantic Matching Energy(SME) [3, 2] introduces two definitions of semantic matching energy functions for optimization, a linear form $f(h, r, t) = (\mathbf{M}_1\mathbf{h} + \mathbf{M}_2\mathbf{r} + \mathbf{b}_1)^T(\mathbf{M}_3\mathbf{t}+\mathbf{M}_4\mathbf{r}+\mathbf{b}_2)$, and a bilinear form $f(h, r, t) = (\mathbf{M}_1\mathbf{h}\otimes\mathbf{M}_2\mathbf{r}+\mathbf{b}_1)^T(\mathbf{M}_3\mathbf{t}\otimes\mathbf{M}_4\mathbf{r} + \mathbf{b}_2)$, where $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4$ are weight matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias vectors and $\otimes$ is Hadamard product.

Single Layer Model(SLM) [17] is designed as a plain baseline of NTN. It introduces nonlinear transformations by neural networks. The score function is $f(h, r, t) = \mathbf{u}_r{}^T g(\mathbf{M}_{rh}\mathbf{h} + \mathbf{M}_{rt}\mathbf{t} + \mathbf{b}_r)$, where $\mathbf{M}_{rh}$ and $\mathbf{M}_{rt}$ are weight matrices, and $g(\cdot)$ is the function $\tanh(\cdot)$.

The Neural Tensor Network(NTN) [17] uses a bilinear tensor layer related two entity vectors to replace a standard linear neural network layer. It computes a score to measure the plausibility of a triplet (h, r, t) by the function $f(h, r, t) = \mathbf{u}_r{}^T g(\mathbf{h}^T\mathbf{M}_r\mathbf{t} + \mathbf{V}_r[\mathbf{h}; \mathbf{t}] + \mathbf{b}_r)$ where $g(\cdot) = \tanh(\cdot)$; $[\mathbf{h}; \mathbf{t}]$ denotes the vertical stacking of vectors $\mathbf{h}$ and $\mathbf{t}$, $\mathbf{V}_r$ is weight matrix and $\mathbf{M}_r$ is a 3-way tensor.

3 Our Model

TransGH considers the translation operation on a generalized hyperplane determined by a set of basis vectors, to achieve the generalized ability for preserving mapping properties of complex relation facts, and also avoid much more computations on entity mappings.

3.1 Generalized Hyperplane

We extend the hyperplane of TransH to the generalized hperplane by a set of basis vectors $\{\mathbf{w}_r^1, \mathbf{w}_r^2, ..., \mathbf{w}_r^v\}$, $(\mathbf{w}_r^i \in R^m, i \in [1, v])$, the basis vectors are orthogonal to each other. With the same setting of TransH, we also restrict $\|\mathbf{w}_r^i\|_2 = 1$ for each set

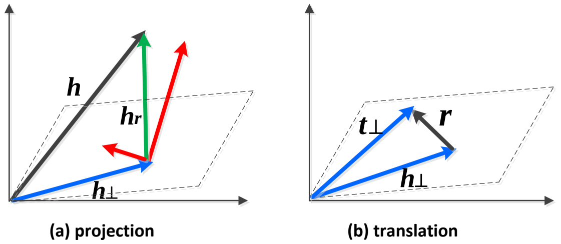


Fig. 2. The two phases of TransGH. The red bold arrows represent $\mathbf{w}_r^{i\,T}\mathbf{h}\mathbf{w}_r^i$.

of r-relation vectors. For an entity embedding $\mathbf{e}$, a transfer vector $\mathbf{e}_r$ on the set of basis vectors can be written as:

$$\mathbf{e}_r = \mathbf{w}_r^{1\,T}\mathbf{e}\mathbf{w}_r^1 + \ldots + \mathbf{w}_r^{v\,T}\mathbf{e}\mathbf{w}_r^v = \sum_i \mathbf{w}_r^{i\,T}\mathbf{e}\mathbf{w}_r^i$$

where v is the number of vectors and m is the dimension of entity (relation) vector space. Based on the transfer vector $\mathbf{e}_r$, we can obtain the projection $\mathbf{e}_\perp$ of entity embedding $\mathbf{e}$ on the generalized hyperplane as $\mathbf{e}_\perp = \mathbf{e} - \mathbf{e}_r$. Thus the generalized hyperplane determined by the set of basis vectors $\{\mathbf{w}_r^1, \mathbf{w}_r^2, ..., \mathbf{w}_r^v\}$, can be described as

$$\{\mathbf{e}_\perp | \mathbf{e}_\perp = \mathbf{e} - \sum_i \mathbf{w}_r^{i\,T}\mathbf{e}\mathbf{w}_r^i\}$$

where $\mathbf{w}_r^i \in R^m$ and $\|\mathbf{w}_r^i\|_2 = 1$. The proposed hyperplane is a generalisation of that in TransH.

3.2 TransGH

As shown in Figure 2, the basic idea of TransGH can be summed up in two steps: (1) *projection*: projecting entity embeddings on the generalized hyperplane.(2) *translation*: connecting projected entities with the relation-specific translation vector. Specifically, for a triplet (h, r, t):

• In the *projection* phase, with the restriction $\|\mathbf{w}_r^i\|_2 = 1$, it is easily to get the projections of head and tail embedding on the generalized hyperplane, that is

$$\mathbf{h}_\perp = \mathbf{h} - \sum_i \mathbf{w}_r^{i\,T}\mathbf{h}\mathbf{w}_r^i, \quad \mathbf{t}_\perp = \mathbf{t} - \sum_i \mathbf{w}_r^{i\,T}\mathbf{t}\mathbf{w}_r^i$$

• In the *translation* phase, the relation r is interpreted as the translation vector $\mathbf{r}$ from the head projections $\mathbf{h}_\perp$ to the tail projection $\mathbf{t}_\perp$. Therefore, the score function is denoted as:

$$f(h, r, t) = \|(\mathbf{h} - \sum_i \mathbf{w}_r^{i\,T}\mathbf{h}\mathbf{w}_r^i) + \mathbf{r} - (\mathbf{t} - \sum_i \mathbf{w}_r^{i\,T}\mathbf{t}\mathbf{w}_r^i)\|_2^2$$

The score function is to measure the compatible of a positive triplet, and also is expected to be low for a positive triplet, otherwise high for a negative triplet.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_48

3.3 Training Method and Implementation Details

We use the following margin-based loss function to encourage discrimination between positive triplets and negative triplets:

$$\text{Ł} = \sum_{(h,r,t)\in P} \sum_{(h',r,t')\in N} [0, f(h,r,t) + \gamma - f(h',r,t')]_+$$

Here, $[x]_+ = max(0,x)$ means to get the maximum number between 0 and x, P is the set of positive triplets; N is the set of negative triplets, that is $N = \{(h',r,t) \mid (h' \in \mathtt{E} \wedge h' \neq h) \cup (h,r,t') \mid (t' \in \mathtt{E} \wedge t' \neq t)\}$. $\mathtt{E}$ is the entities set. $\gamma > 0$ is the margin hyper-parameter with expectation of dividing the positive triplets and negative triplets. Then we minimize the loss function with considering the following constraints:

$$\forall e \in \mathtt{E}, \|\mathbf{e}\|_2 \le 1, \forall r \in \mathtt{R}, \|\mathbf{r}\|_2 \le 1 \tag{1}$$

$$\forall r \in \mathtt{R}, i \in [1,v], \|\mathbf{w}_r^i\|_2 = 1 \tag{2}$$

$$\forall r \in \mathtt{R}, i \in [1,v], \frac{|\sum_i {\mathbf{w}_r^i}^T \mathbf{r}|}{\|\mathbf{r}\|_2} \le \epsilon \tag{3}$$

$$\forall r \in \mathtt{R}, i,j \in [1,v](i \neq j), \frac{|\sum_{(i,j)} {\mathbf{w}_r^i}^T \mathbf{w}_r^j|}{\|\mathbf{w}_r^j\|_2} \le \epsilon \tag{4}$$

where ϵ is a small scalar, $\mathtt{R}$ is the relations set, constraint (3) assures the translation vector $\mathbf{r}$ is on the generalized hyperplane and constraint (4) guarantees each two basis vectors are orthogonal. Afterwards we directly optimize the following loss function with soft constraints:

$$\begin{aligned}\text{Ł} = &\sum_{(h,r,t)\in P} \sum_{(h',r,t')\in N} [0, f(h,r,t) + \gamma - f(h',r,t')]_+ \\ &+ C(A_1 + A_2)\end{aligned} \tag{5}$$

where we set

$$\begin{aligned} A_1 &= \sum_{e\in\mathtt{E}} [\|\mathbf{e}\|_2^2 - 1]_+ + \sum_{r\in\mathtt{R}} [\|\mathbf{r}\|_2^2 - 1]_+ \\ A_2 &= \sum_{r\in\mathtt{R}} \{[(\frac{\sum_i {\mathbf{w}_r^i}^T \mathbf{r}}{\|\mathbf{r}\|_2})^2 - \epsilon^2]_+ + [(\frac{\sum_{(i,j)} {\mathbf{w}_r^i}^T \mathbf{w}_r^j}{\|\mathbf{w}_r^j\|_2})^2 - \epsilon^2]_+\} \end{aligned} \tag{6}$$

and C is a hyper-parameter used to measure the importance of soft constrains.

The loss function favors the lower scores for positive triplets than that for negative triplets. We adopt stochastic gradient descent(SGD) [8] to minimize the above loss function. Notice that the constrain (2) is missed in Eq 5. To satisfy it, we set each vector $\mathbf{w}_r^i$ to unit l_2-ball before traversing each mini-batch. Moreover, negative triplets are generated via replacing either the head or tail of original triplets exited in KGs by a random entity, but not both at the same time. For reducing the false negative triplets, here we follow [19] and set different probabilities for the replacement. In experiment, the traditional sampling method is denoted as ”unif” and the new method [19] as ”bern”.

Table 1. Complexity(the number of parameters and the number of multiplication operations).

Model	# Parameters	# Operations(Time complexity)
UM [4]	$O(N_em)$	$O(N_t)$
SE [6]	$O(N_em + 2N_rn^2)(m = n)$	$O(2m^2N_t)$
LFM [10]	$O(N_em + N_rn^2)(m = n)$	$O((m^2 + m)N_t)$
SME(BILIN) [2]	$O(N_em + N_rn + 4mks + 4k)(m = n)$	$O(4mksN_t)$
SLM [17]	$O(N_em + N_r(2k + 2nk))(m = n)$	$O((2mk + k)N_t)$
NTN [17]	$O(N_em + N_r(n^2s + 2ns + 2s))(m = n)$	$O(((m^2 + m)s + 2mk + k)N_t)$
TransE [5]	$O(N_em + N_rn)(m = n)$	$O(N_t)$
TransH [19]	$O(N_em + 2N_rn)(m = n)$	$O(2mN_t)$
TransR [12]	$O(N_em + N_r(m + 1)n)$	$O(2mnN_t)$
TransD [11]	$O(2N_em + 2N_rn)$	$O(2nN_t)$
TransE-RS [20]	$O(N_em + N_rn)(m = n)$	$O(N_t)$
TransH-RS [20]	$O(N_em + 2N_rn)(m = n)$	$O(2mN_t)$
TransGH (this paper)	$O(N_em + N_r(1 + v)n)(m = n), v \ll m$	$O(2vmN_t)$

Generally, all embeddings of entities $\{\mathbf{e}_i\}_{i=1}^{|\mathbb{E}|}$, relations $\{\mathbf{r}_k\}_{k=1}^{|\mathbb{R}|}$ and relation-specific vectors $\{\mathbf{w}_r^1, \mathbf{w}_r^2, ..., \mathbf{w}_r^v\}_{r=1}^{|\mathbb{R}|}$ are learned by TransGH. Hence parameters of this model is $N_em + N_r(1+v)n$ and the time complexity is $2vmN_t$, which is similar to TransH as we usually set $v \ll m$, e.g., $v = 2, 3, 4$. We compare the parameters and time complexities with several baselines in Table 1. We denote N_e as the number of entities, N_r as the number of relations and N_t as the number of triplets in a knowledge graph respectively. m and n separately represent the dimension of entity space and relation space. d denotes the average number of clusters of a relation. k is the number of hidden nodes of a neural network, s is the number of slice of a tensor. v is the number of vectors for a relation.

4 Experiments and Analysis

We study and evaluate our model on two tasks: link prediction [5, 19] and triplet classification [17]. In our experiments, two datasets including FreeBase [1] and WordNet [15] are used. Then we show the experimental results and some analysis of them.

4.1 Datasets

WordNet is designed to build an usable dictionary and support automatic text analysis. In WordNet, each entity represents a *synset* containing several words, which are corresponding to a distinct word sense. Relationships indicate the lexical relations between synsets, such as *hypernym*, *hyponym*, *meronym* and *holonym*. An example of triplets is (*__warship_NN_1*, *_hyponym*, *__torpedo_boat_NN_1*). The two data sets from WordNet, WN18 and WN11, are used in our experiments. WN18 contains 18 relations and WN11 contains 11 relations. The number of entities involved in the two data sets is close.

FreeBase is a large and rising knowledge graph of general facts. An example of FreeBase is (*nietzchka_keene*, *place_of_death*, *madison*), it builds a relation *place_of_death* between a name entity *nietzchka_keene* and a place entity *madison*. We use two data sets with FreeBase in this paper, FB15k and FB13. FB15k consists of 592,213 triplets with 14,951 entities and 1,345 relations. FB13 is a more dense subgraph including 75,043 entities and 13 relations. The statistics of these data sets are listed in Table 2.

Table 2. Data sets used in the experiments.

DataSet	***#Relation***	***#Entity***	***#Train***	***#Valid***	***#Test***
FB15k	1,345	14,951	483,142	50,000	59,071
WN18	18	40,943	141,442	5,000	5,000
FB13	13	75,043	316,232	5,908	23,733
WN11	11	38,696	112,581	2,609	10,544

4.2 Link prediction

Link prediction is to predict the missing h or t for a positive triplet (h, r, t), used in [5, 19, 12, 11]. In this task, it focuses more on ranking a set of candidate entities from the knowledge graph rather than obtaining the best one for each position of missing entity. The data sets used in this task are WN18 and FB15k, which are same settings to [5, 19, 12, 11].

Evaluation Rules. We adopt the same protocols used in [5, 19, 12, 11] to evaluate this task. Specifically, in testing phase, for each test triplet (h, r, t), we replace the head(tail) entity by every entity e from the set of entities for a KG and calculate the scores of these corrupted triplets by using the score function $f(h, r, t)$, then we get the rank of the original triplet after ranking these scores in ascending order. Following [5, 19, 12, 11], two metrics are used to evaluation: the average rank(Mean Rank) and the proportion of ranks not larger than 10 (Hit@10). This is called ”raw” setting. Notice that the corrupted triplets may exit in the KG, they can be regarded as correct triplets, hence it is not wrong to rank them before the original triplet. To eliminate this case, we filter out corrupted triplets existing in a KG before ranking. This is called ”filt” setting. In both settings, lower Mean and higher Hit@10 are excepted.

Implementation. In training phase, we select the learning rate η for SGD from $\{$0.001, 0.01, 0.1$\}$, the γ from$\{$1, 2, 3, 4, 5, 6, 7, 8$\}$, the entity(relation) embedding dimension m from$\{$50, 100, 150$\}$, the number of vectors v from $\{$0.25, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$\}$, the batch size b from$\{$480, 960, 1200, 4800$\}$, the hyper-parameter C from $\{$0.005, 0.0625, 0.25, 0.5$\}$. The best parameters are determined by validation set. Under *unif* setting, the best optimal configures are η = 0.01, γ = 7, m = 100, v = 2, b = 1200, C = 0.0625 on WN18; η = 0.01, γ = 2, m = 100, v = 6, b = 1200, C = 0.0625 on FB15k. Under *bern* setting, the best optimal configures are η = 0.01, γ = 7, m = 100, v = 2, b = 1200, C = 0.005 on WN18; η = 0.01, γ = 1, m = 100, v = 4, b = 480, C = 0.0625 on FB15k. We traverse all the training triplets for 5000 rounds and take *L1* as dissimilarity on both datasets.

Table 3. Link prediction results.

Dataset	***WN18***				***FB15k***			
Metric	Mean		Hits@10		Mean		Hits@10	
	raw	filt	raw	filt	raw	filt	raw	filt
RESCAL [16]	1,180	1,163	37.2	52.8	828	683	28.4	44.1
UM [4]	315	304	35.3	38.2	1074	979	4.5	6.3
SE [6]	1011	985	68.5	80.5	273	162	28.8	39.8
SME(LIN) [3]	545	533	65.1	74.1	274	154	30.7	40.8
SME(BILIN) [2]	526	509	54.7	61.3	284	158	31.3	41.3
BILINEAR [10]	469	456	71.4	81.6	283	164	26.0	33.1
TransE [5]	263	251	75.4	89.2	243	125	34.9	47.1
TransH(unif) [19]	318	303	75.4	86.7	211	84	42.5	58.5
TransH(bern) [19]	400.8	388	73.0	82.3	212	87	45.7	64.4
TransR(unif) [12]	232	219	78.3	91.7	226	78	43.8	65.5
TransR(bern) [12]	238	225	79.8	92.0	198	77	48.2	68.7
CTransR(unif) [12]	243	230	78.9	92.3	233	82	44	66.3
CTransR(bern) [12]	231	218	79.4	92.3	199	75	48.4	70.2
TransD(unif) [11]	242	229	79.2	92.5	211	67	49.4	74.2
TransD(bern) [11]	224	212	79.6	92.2	194	91	53.4	77.3
TransE-RS(unif)[20]	362	348	80.3	93.7	**161**	**62**	53.1	72.3
TransE-RS(bern)[20]	385	371	80.4	93.7	**161**	**63**	53.2	72.1
TransH-RS(unif)[20]	401	389	81.2	94.7	163	64	53.4	72.6
TransH-RS(bern)[20]	371	357	80.3	94.5	178	77	53.6	75.0
TransGH(unif)	**191**	**179**	**81.4**	**94.8**	186	66	**54.0**	**79.8**
TransGH(bern)	**210**	**197**	**81.6**	**95.3**	186	64	**54.1**	**80.1**

Results. The results on both WN18 and FB15k are shown in Table 3. The results of previous studies are referred from their report, since the same datasets are used. Our model consistently and significantly outperforms previous models on both the metrics of WN18 and FB15k, where the results of our Mean(raw) is 191, Mean(filt) is 179, Hit@10(raw) is 94.8%, Hit@10(filt) is 95.0% on WN18, and that of Mean(raw) is 186, Mean(filt) is 64, Hit@10(raw) is 54.1% and Hit@10(filt) is 80.1% on FB15k. Moreover, our model has respectively remarkable improvements on metrics of Mean(raw), Mean(filt), Hit@10(raw) and Hit@10(filt) comparing with TransH, which are 172, 124, 6.2% and 8.3% on WN18, and 25, 23, 8.4% and 15.7% on FB15k higher than those of TransH. We believe the improved performance of our model is due to its use of the set of basis vectors.

Table 4 analyzes Hits@10 results on FB15k with respect to the relation categories. Following the same rules in [5] on FB15k, we separate the 1345 relations into four categories, including one-to-one, one-to-many, many-to-one and many-to-many relations. From Table 4 we can observe that TransGH significantly performs better results than all baselines on both *unif* and *bern* settings. Our method has highest accuracies on predicting head(one-to-one 87.0%, one-to-many 95.8%, many-to-one 47.9% and many-to-many 80.8%) and predicting tail(one-to-one 86.8%,one-to-many 55.8%, many-to-one 94.8% and many-to-many 84.3%). Additionally, comparing with TransH, we also give the result on Hit@10 metric of some typical complex relations in Table 5. In this experiment, we directly copy the results reported in [19]. shows TransGH has remark-

Table 4. Results on FB15k by relation category.

Dataset	***Predicting left (Hit@10)***				***Predicting right (Hit@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
UM [6]	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE [6]	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME(LIN) [3]	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME(BILIN) [2]	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE [5]	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH(unif) [19]	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH(bern) [19]	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR(unif) [12]	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransR(bern) [12]	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
CTransR(unif) [12]	78.6	77.8	36.4	68.0	77.4	37.8	78.0	70.3
CTransR(bern) [12]	81.5	89.0	34.7	71.2	80.8	38.6	90.1	73.8
TransD(unif) [11]	80.7	85.8	47.1	75.6	80.0	54.5	80.7	77.9
TransD(bern) [11]	86.1	95.5	39.8	78.5	85.4	50.6	94.4	81.2
TransE-RS(unif) [20]	87.2	**96.2**	35.9	71.8	**87.0**	45.0	**95.5**	75.4
TransE-RS(bern) [20]	87.4	**96.3**	35.3	71.7	86.5	44.2	95.4	75.2
TransH-RS(unif) [20]	87.6	95.9	35.6	72.5	86.3	44.9	**95.5**	75.8
TransH-RS(bern) [20]	85.6	95.5	37.4	75.5	85.7	47.4	94.9	78.7
TransGH(unif)	**86.4**	**95.6**	**47.6**	**80.6**	**85.8**	**55.8**	**94.8**	**83.4**
TransGH(bern)	**87.0**	**95.8**	**47.9**	**80.8**	**86.8**	**55.7**	94.8	**84.3**

able improvement on Hit@10 metric of some typical complex relations compared with TransH. It indicates TransGH can capture more fertile information between entities and relations, and achieve the better ability for modeling mapping properties of complex relation facts. As Table Table 6 and 7 shown, TransGH rationality enables the same category objects(entities and relations) to have similar vector embeddings.

Table 5. Hits@10(filt)*bern* of TransGH and TransH on some examples of one-to-many*, many-to-one†, many-to-many‡ and symmetric§ relations.

Relations	***Hit@10(TransH/TranGH) on FB15k***	
	Predict Head	***Predict Tail***
/football_position/players*	100 / 100	22.2 / **88.9**
/production_company/films*	85.6 / **96.8**	16.0 / **52.4**
/director/film*	89.6 / **96.2**	80.2 / **94.3**
/disease/treatments†	66.6 / 66.6	100 / 100
/person/place_of_birth†	37.5 / **77.9**	87.6 / **92.0**
/film/production_companies†	21.0 / **47.5**	87.8 / **96.7**
/field_of_study/students_majoring‡	66.0 / **92.2**	62.3 / **70.5**
/award_winner/awards_won‡	87.5 / **99.0**	86.6 / **99.5**
/sports_position/players‡	100 / 100	86.2 / **99.6**
/person/sibling_s§	63.2 / **68.4**	36.8 / **68.4**
/person/spouse_s§	35.2 / **70.4**	42.6 / **59.3**

Table 6. The Top-3 similarity entities with regard to some examples on WN18. The similarity scores are computed with *cosine* function.

Dataset	***WN18***	
Entity and Definitions	__mountain_sheep_NN_1	any wild sheep inhabiting mountainous regions
Similar Entities and Definitions	__white_sheep_NN_1	large white wild sheep of northwestern Canada and Alaska
	__rocky_mountain_sheep_NN_1	wild sheep of mountainous regions of western North America having massive curled horns
	__wild_sheep_NN_1	undomesticated sheep
Entity and Definitions	__sharpen_VB_8	make (one's senses) more acute
Similar Entities and Definitions	__screw_up_VB_1	make more intense
	__raise_VB_13	increase
	__intensify_VB_2	make more intense, stronger, or more marked

Table 7. The Top-3 similarity relations with regard to some examples on FB15k. The similarity scores are computed with *cosine* function.

Dataset	***FB15k***
Relation	/location/statistical_region/rent50_3./measurement_unit/dated_money_value/currency
Similar relations	/location/statistical_region/rent50_0./measurement_unit/dated_money_value/currency
	/location/statistical_region/rent50_1./measurement_unit/dated_money_value/currency
	/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency
Relation	/people/person/nationality
Similar relations	/people/person/places_lived./people/place_lived/location
	/people/person/place_of_birth
	/people/deceased_person/place_of_death

4.3 Triplet Classification

Triplet classification is to decide whether a given triplet (h, r, t) is correct or not. This is a binary classification task, which has been presented by [17]. In this task, three data sets WN11, FB13 and FB15k are used, and negative triplets are needed to the evaluation of binary classification. The first two sets appeared in [17] already have negative triplets, but the third one including negative triplets has not been published recently. For FB15k, we construct it by following the same principles used for FB13 in [17].

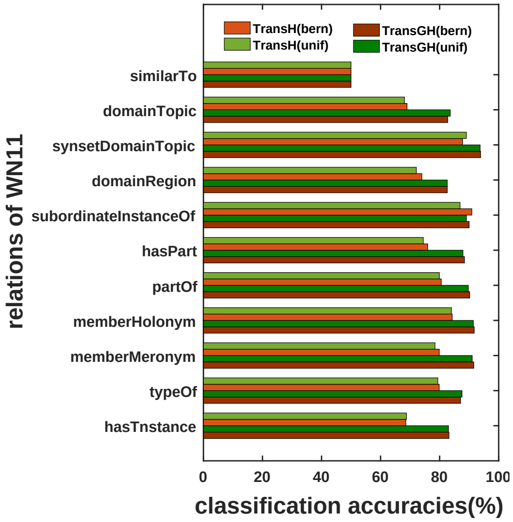
Evaluation Rules. There exists a simple decision rule for triplet classification: we first get a relation-specific threshold δ_r determined by maximizing the classification accuracy on the validation set. For a triplet (h, r, t), if the dissimilarity score gained by the score function $f(h, r, t)$ is below δ_r, then predict positive. Otherwise predict negative.

Implementation. We compare our model with several baseline methods mentioned in [11]. For the sake of fairness, word embedding [14] is not used in our experiments. In training stage, we select the same configuration with link prediction. The best parameters are also determined by validation set. On *unif* setting, the best optimal configures are η = 0.01, γ = 11, m = 100, v = 3, b = 480, C = 0.25 on WN11; η = 0.01, γ = 0.25, m = 100, v = 2, b = 1200, C = 0.0625 on FB13; η = 0.01, γ = 1, m = 100, v = 6, b = 480, C = 0.0625 on FB15k. On *bern* setting, the best optimal configures are η = 0.01, γ = 11, m = 100, v = 3, b = 480, C = 0.0625 on WN11; η = 0.01, γ = 0.25, m = 100, v = 2, b = 1200, C = 0.005 on FB13; η = 0.01, γ = 1, m = 100, v = 10, b = 480, C = 0.0625 on FB15k. We set the number of epochs to 5000 for three data sets. Meanwhile we also take *L1* as dissimilarity on WN11, FB15k and *L2* on FB13.

Table 8. triplet classification accuracies.

Dataset	***WN11***	***FB13***	***FB15k***
SLM	69.9	85.3	-
NTN	70.4	87.1	-
SE	53.0	75.2	72.2
SME	70.0	63.7	71.6
TransE(unif)	75.9	70.9	79.5
TransE(bern)	75.9	81.5	80.4
TransH(unif)	77.7	76.5	79.9
TransH(bern)	78.8	83.3	80.0
TransR(unif)	85.5	74.7	81.2
TransR(bern)	85.9	82.5	82.5
TransD(unif)	85.6	85.9	86.0
TransD(bern)	86.4	89.1	88.2
TransE-RS	85.3	83.0	81.9
TransH-RS	86.4	81.6	83.2
TransGH(unif)	**87.2**	84.7	**91.4**
TransGH(bern)	**87.3**	85.2	**91.2**

Fig. 3. Classification accuracies of on WN11.



Results. Evaluation results of triplet classification are shown in Table 8. TransGH consistently scores better accuracy on WN11 and FB15k than the current state-of-the-art model, where accuracies are 87.3% and 91.2% on WN11 and FB15k respectively. TransGH has slightly worse accuracy on FB13. This is mainly because that FB13 has the most entities and therefore good representations of rarely occurring entities are difficult for learning. Additionally TransGH achieves at least 8.5%, 1.9%, 11.4% higher than TransH on the three datasets. Therefore we believe the set of basis vectors is beneficial to model the complex relations and learn the embeddings of entities and relations of a knowledge graph. We also compare the classification accuracies of different relations by TransH and TransGH on WN11. In this experiment, we rerun TransH with the parameters reported in [19], and obtain slightly different accuracies 76.5%(*unif*) and 77.6%(*bern*) with the reported results in Table 8. We ignore the differences derived from randomly experiments. The accuracies of eleven relations on WN11 are given separately in Figure 3. From results of Figure 3, TransGH significantly improve TransH in each relation classification expect for the relation *_similar_to*. As reported in [11], the prediction accuracy needs more information while the number of entity pairs linked by relation *_similar_to* only accounts for 1.5% in all train data, therefore the inadequate entity pairs linked by relation *_similar_to* is the main cause.

5 Conclusion and Future Work

In this paper, we have proposed a new knowledge graph embedding method TransGH. The key idea of TransGH is to learn embeddings via modeling each relation as the translation vector between projected entities on the generalized hyperplane, which is characterized by a set of basis vectors. In addition, TrasGH is efficient for preserving mapping properties of complex relation facts while keeping low complexity of parameters. We empirically conduct experiments on triplet classification and link prediction with two knowledge graphs FreeBase and WordNet. The experimental results show that TransGH significantly and consistently has considerable improvement over baselines, and achieve state-of-the-art performance, which demonstrates the superiority and generality of our model.

In the future, we will explore the following directions: (1) We will utilize the word embeddings obtained from *word2vec*[13] in our experiments for improving the performance of our model TransGH. (2) We will train our model TransGH using the promising limit-based scoring loss function introduced by [20] for future improvement. (3) We will devise and exploit a question answering system based on TransGH.

Acknowledgment

This work is supported by National Key R&D Program No.2017YFB0803003, and the National Natural Science Foundation of China (No.61202226), We thank all anonymous reviewers for their constructive comments.

References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase:a collaboratively created graph database for structuring human knowledge. In: ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, Bc, Canada, June. pp. 1247–1250 (2008)
2. Bordes, A., , Weston, X.G.J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data. CoRR **abs/1301.3485** (2013)
3. Bordes, A., Glorot, X., Weston, J.: Joint learning of words and meaning representations for open-text semantic parsing. In: International Conference on Artificial Intelligence and Statistics (2011)
4. Bordes, A., Glorot, X., Weston, J.: Joint learning of words and meaning representations for open-text semantic parsing. In: International Conference on Artificial Intelligence and Statistics (2012)
5. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. pp. 2787–2795 (2013)

6. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, Usa, August (2011)
7. Bordes, A., Weston, J., Usunier, N.: Open question answering with weakly supervised embedding models. In: European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 165–180 (2014)
8. Bottou, L.: Large-scale machine learning with stochastic gradient descent pp. 177–186 (2010)
9. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 541–550 (2011)
10. Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. In: International Conference on Neural Information Processing Systems. pp. 3167–3175 (2012)
11. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. pp. 687–696 (2015)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. pp. 2181–2187 (2015)
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. pp. 3111–3119 (2013)
15. Miller, G.A.: Wordnet: a lexical database for english. Communications of the Acm **38**(11), 39–41 (1995)
16. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011. pp. 809–816 (2011)
17. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. pp. 926–934 (2013)
18. Sutskever, I., Salakhutdinov, R., Tenenbaum, J.B.: Modelling relational data using bayesian clustered tensor factorization. In: Advances in Neural Information Processing Systems 22: Conference on Neural Information Processing Systems 2009. Proceedings of A Meeting Held 7-10 December 2009, Vancouver, British Columbia, Canada. pp. 1821–1828 (2009)
19. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada. pp. 1112–1119 (2014)
20. Zhou, X., Zhu, Q., Liu, P., Guo, L.: Learning knowledge embeddings by combining limit-based scoring loss. In: CIKM 2017. pp. 1009–1018 (2017)