CoRelatE: Modeling the Correlation in Multi-fold Relations for Knowledge Graph Embedding

Yan Huang[a], Ke Xu[a], Xinyu Wang[b], Haili Sun[a], Songfeng Lu[a,c,*], Tongyang Wang[a], Xinfang Zhang[a]

[a] *School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China.*
[b] *709th Research Institute, China Shipbuilding Industry Corporation, Wuhan 430074, China.*
[c] *Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen 518063, 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, TransH and TransR 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 demonstrate that our approach outperforms related methods by a significant margin.

Keywords: knowledge base, representation learning, multi-fold relation

*Corresponding author
Email addresses: platanus@hust.edu.cn (Yan Huang), lusongfeng@hust.edu.cn (Songfeng Lu)

1. Introduction

Knowledge bases are directed graphs with nodes representing entities and edges representing relations between entities. Great achievements have been made in building large scale knowledge bases, such as Freebase [1], WordNet [2], YAGO [3] and DBpedia [4]. These knowledge bases can be used in many areas like semantic search, question answering, drug discovery and disease diagnosis. Although the current knowledge bases contain large amounts of entities and relations, they are far from completeness. This calls for knowledge base completion techniques to inference or predict missing entities and unknown links between entities based on existing ones. Furthermore, the entities and relations in the knowledge bases are symbolic and inadequate for inference or calculation.

To this end, representation learning [5] has been proposed as a kind of promising approach for knowledge base completion. 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 entire entities and relations. Compared with the traditional logic-based inference approaches, representation learning shows strong feasibility and robustness in the applications.

Despite the promising achievements, most existing techniques for knowledge base representation learning (such as TransE [6], TransH [7], TransR [8] and ProjE [9], ComplEx [10] etc.) consider only binary relations therein, namely RDF(Resource Description Framework) data with triples each involving two entities and a binary relation between them. For example, "Donald J. Trump is the president of America" consists of two entities "Donald J. Trump", "America" and a binary relation "president_of_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"). As reported[11, 12], more than 1/3rd of the entities in Freebase participate in non-binary relations and 61% of the relations are non-binary.

A general approach for this problem is to convert each multi-fold relation into multiple triples with binary relations and learn the embedding of each triple using the existing translating embedding methods [5, 6, 7, 8]. Thus, an instance with an N-ary relation is converted to $\binom{N}{2}$ triples according to the S2C [1] conversion[11] or to N triples via RDF reification[12]. Although S2C conversion is capable of capturing part of the structures of multi-fold relations [13], it's irreversible and leads to a heterogeneity of the predicates, unfavorable for embedding. As to RDF reification method, it's difficult to reify the test samples and define a way to embed the newly created entities when we have little information about them. Wen et al.[11] advocates an instance representation of multi-fold relations and proposes a direct modeling framework "m-TransH" for knowledge base embedding. However, we show that m-TransH has several shortcomings:

(1) Treats fact nodes the same as general entity nodes, this can be seen from the loss function of m-TransH:ID[2] in which embeddings of entities and facts of the same N-ary relations are calculated via a linear combination operator. Thus do harm to the learning of semantic information and to distinguishing of these two types of nodes.

(2) For binary relations, the relation between two entities can be easily estimated via cost function of most existing knowledge base embedding models (e.g. $\|h + r - t\|_2^2$ from TransE[6] model). While for multi-fold relations, the m-TransH:ID model, we can hardly infer or classify the relation type of a set of entities as an instance without the fact node information.

(3) The correlation of facts and their linked entities can not be explicitly

[1] Star-to-Clique Conversion: For each two entities (with relation role r1 and r2) of an N-ary relation instance, form a labeled edge r1.r2 between them, then delete the instance node and all edges (i.e. relations) connecting to it.

[2] m-TransH:ID is a variation of m-TransH for multi-fold relational instances with fact nodes in FACT-ID role.

learned through linear combination of their embeddings, which we argue to be a key factor for multi-fold relation embedding.

(4) Ignores the associated information of facts and relations that certain facts belong to the same relation. It's impossible to infer the relation type of a fact through the fact embedding only and to discover the facts of the same relation type.

In this paper, we first discuss the problems of existing models for multi-fold relation embedding, why they are important, our motivation, necessity and significance.

Then we extend the m-TransH model and present a Group-constrained Embedding method which embeds fact nodes as well as entity nodes and their relations into three different vector space (named as entity embedding space, fact embedding space and relation embedding space). We model three correlations, namely, correlation between entities and their relations, correlation between entities and facts, correlation between facts and their relation types. For the first correlation, we extend the m-TransH model by replacing the coefficient function $a_r(\rho)$ of entity embeddings $\mathbb{P}_{n_r}(t(\rho))$ to a diagonal matrix $M_r(\rho)$ to improve it's expressive ability. To model the correlation between entities and facts, we utilize a graph convolutional network to learn the embedding of facts through the connected entities. Then we restrict the embedded fact nodes related to the same relation to groups with three different constraint strategies, i.e. zero constraint (by making the embedded fact vector to be close to its corresponding relation vector indefinitely), radius constraint (by forcing the Euclidean distance between the embedded fact vector and its corresponding relation vector to be smaller than a radius ϵ) or cosine constraint (by rendering the cosine distance between the embedded fact vector and its corresponding relation vector to be zero).

The Group-constrained Embedding method for knowledge base with multi-fold relations is named as "Gm-TransH" . 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 [6] and JF17K [11]. 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 by a large margin and achieves up to 13.8% improvement over the comparative models.

The main contributions of our work are as follows:

(a) Present a Group-constrained Embedding method for multi-fold relation embedding (Gm-TransH), which not only embed entities and relations into continuous vector space, we also learn the embedding of facts and model the correlation with entities and relations explicitly.

(b) Instead of modeling the correlations of entities, facts and relations as a whole, we learn the correlation of each two of them separately. This settles the 4 shortcomings of m-TransH and is shown to be efficient for predicting and classification.

(c) To model the correlation between facts and relation types, we constrain the instantiated fact embeddings to be close to their belonging relation embeddings and far away from others that are not. In this way, the translation embedding model for entity-relation correlation learning and the GCN model for entity-fact correlation learning are interacted and balanced during the training stage.

(d) We clean the redundant data and generate a new subset G_{fact} for the JF17K datasets in which the repetitive instances are removed and the missing facts are appended.

(e) We compare our model with the optimal translating embedding approaches, tensor factorization approaches and the most up to dated GCN models on several canonical datasets and have shown continuous improvements over the existing models.

2. Motivation and Related Work

The most related works on representation learning of knowledge bases can be divided into two classes: binary relation embedding and multi-fold relation embedding. These works are briefly summarized in Table 1.

2.1. Binary Relation Embedding

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

TransE [6] sets $(h + r)$ to be the nearest neighbor of t when (h, r, t) holds, far away otherwise. The cost function is defined as

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

TransH [7] is developed to enable an entity to have distinct distributed representations when involved in different relations. For a relation r, TransH models the relation as a vector r on a hyperplane with n_r as the normal vector. For a triple (h, r, t), the entity embeddings h and t are first projected to the hyperplane of n_r, denoted as $h_\perp$ and $t_\perp$. The cost function is defined as

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

where $h_\perp = n_r^T h n_r$ and $t_\perp = n_r^T t n_r$.

TransR [8] models entities and relations in distinct spaces and performs translation in relation space. For each relation r, a projection matrix M_r is used to project entities from entity space to relation space, i.e. $h_r = hM_r$, $t_r = tM_r$. The cost function is correspondingly defined as

$$f_r(h, t) = \|h_r + r - t_r\|_2^2 \tag{3}$$

Besides TransE, TransH and TransR, there are also many other embedding methods based on binary relations, such as Unstructured Model(UM) [14], Structured Embedding Model(SME) [15], Single Layer Model(SLM) [16], Semantic Matching Energy Model(SME) [17, 14] and Neural Tensor Network Model(NTN) [16], Latent Factor Model(LFM) [18], PTransE [5], TransA [19], TransD [20], TranSparse [21], KG2E [22], ITransF [23], ProjE [9] and so on.

2.2. Multi-fold Relation Embedding

For knowledge bases with multi-fold relations, S2C conversion and decomposition framework [11] are usually used. Then, multi-fold relations are converted to triples and treated as binary relations.

We find that the existing models for multi-fold relation embedding directly without converting into binaries mostly focus on modeling either the correlation between entities and their relations or the relatedness of entities participate in a common instance. These methods neglect the fact information and it's relatedness to entity components and relation types.

Wen et al. [11] 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

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

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

$$\mathbb{P}_{n_r}(z) = z - n_r^\top z n_r \tag{5}$$

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

$$\sum_{\rho \in \mathcal{M}(R_r)} a_r(\rho) = 0 \tag{6}$$

3. Representation and Embedding Problem Definition

From an algebraic point of view, a multi-fold (n-ary) relation on set $\mathcal{N}$ is defined as a subset of the cartesian product of n sets $\mathcal{N} \times \mathcal{N} \times \cdots \times \mathcal{N}$, namely

$\mathcal{N}^n$. Each coordinate of the n-dimensional cartesian product should be specified to a different role of the relation.

We focus on the representation and embedding of multi-fold relations in the form of either triples (with binary relations), instances (with n-ary relations, $n \geq 2$) or facts (each role may involves a list of entities) in knowledge bases. We first introduce an unified way to represent knowledge base (KB) with multi-fold relations, then based on the unified representation, we discuss the embedding problem of these n-ary relations and give a mathematical definition of the problem.

Unified Relation Representation

Different from the most common methods to represent KB as a collection of entity nodes, relations and samples of triples or instances, we create a fact node (a.k.a. instance node) to represent the relation instance with links to all its participants (i.e. entity nodes) and employ an unified framework to represent KB as a collection of entity nodes, fact nodes, relations and relation instances. We design the unified representation framework as below:

For a given knowledge base (KB) $\mathcal{G}$, let $\mathcal{N}_e$ denote the set of all entities in $\mathcal{G}$, $\mathcal{R}$ indexes a set of distinct *multi-fold relations* on $\mathcal{N}_e$, $\mathcal{T}$ denotes a set of instances defined over $\mathcal{N}_e$ and $\mathcal{R}$. We create a fact (or instance) node for each instance in $\mathcal{T}$ and form a fact node set $\mathcal{N}_f$. For each index $r \in \mathcal{R}$, relation R_r on entities $\mathcal{N}_e$ with roles $\mathcal{M}(R_r)$ is a subset of $\mathcal{N}_e^{\mathcal{M}(R_r)}$, where $\mathcal{M}(R_r)$ is a set of ordered role tuples $\{\rho_1, \rho_2, ..., \rho_{|\mathcal{M}(R_r)|}\}$ of relation R_r, $\mathcal{N}_e^{\mathcal{M}(R_r)}$ denotes the set of all functions mapping from $\mathcal{M}(R_r)$ to $\mathcal{N}_e$. We call R_r a J-fold or J-ary relation if cardinality $|\mathcal{M}(R_r)| = J$, J is the "fold" or "arity" of R_r, each relation R_r is allowed to have an arbitrary arity. Let $\mathcal{T}_r$ be the set of instances of relation R_r in $\mathcal{G}$, each instance $t \in \mathcal{T}_r$ in the relation R_r is a vector $(x_1, x_2, ..., x_{|\mathcal{M}(R_r)|})$ of entities, in which entity x_i corresponds to role $\rho_i \in \mathcal{M}(R_r)$, and instance t corresponds to a unique fact node $u_j \in \mathcal{N}_f$. The relation R_r can be semantically understood from the set of all such participated vectors. Then the knowledge base $\mathcal{G}$ can be specified as $(\mathcal{N}_e, \mathcal{N}_f, \mathcal{R}, \{\mathcal{T}_r : r \in \mathcal{R}\})$, named as *unified relation representation.*

We argue that this unified relation representation method can represent a variety of relations, including binary relations as triples, n-ary relations ($n \geq 2$) as instances and relations (such as facts) whose roles may involve a list of entities.

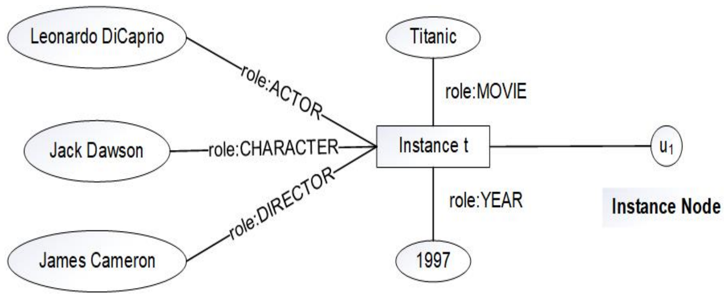

Figure 1: Illustration of unified relation representation for an instance of 5-ary relation "who played which role in which film directed by whom in which year".

For example, as shown in Figure 1, instance *"Leonardo DiCaprio played the role of Jack Dawson in the film 'Titanic' directed by James Cameron in 1997"* should be written as *t=(Leonardo DiCaprio, Jack Dawson, Titanic, James Cameron, 1997)*, which involves 5 entities each corresponding to a role in 5-fold relation $\mathcal{M}(R_r)$:={ACTOR, CHARACTER, MOVIE, DIRECTOR, YEAR}. An instance node u_1 is attached to the instance to represent its structural information and enable the learning of correlation of instances with entities and relations in embedding stage below.

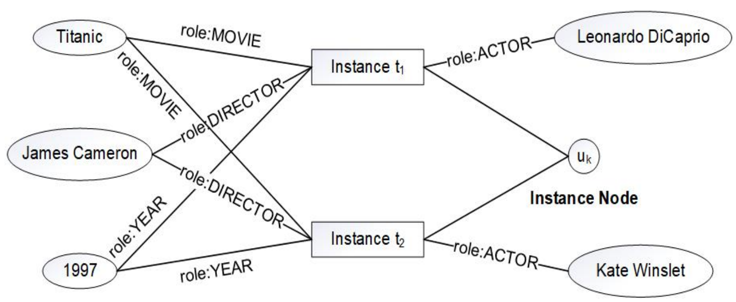

Figure 2: Illustration of unified relation representation for fact with role 'ACTOR' involve two entities 'Leonardo DiCaprio' and 'Kate Winslet'. This fact is converted into two instances, i.e. t_1 and t_2, but share the same fact node u_k.

For multi-fold relations in the form of facts, where each role (or participant) $\rho_i \in \mathcal{M}(R_r)$ of the relation R_r may involve an ordered list of individuals (i.e. entities) rather than a single individual x_i. We follow *m-TransH* and convert the multi-fold relations from a fact to several instances, i.e. *instance representation*, in which each role ρ_i corresponds to an unique entity x_i. Meanwhile, we introduce a set of *fact node*, denoted as $\mathcal{N}_f$, instances of the same fact share a same fact node, this information is shown to be useful in *m-TransH*. For example, in Figure 2, a fact *"Leonardo DiCaprio and Kate Winslet acted in the film 'Titanic' directed by James Cameron in 1997"* can be converted into two instances of 4-fold relation "who acted in which film directed by whom in which year", namely *t1=(Leonardo DiCaprio, Titanic, James Cameron, 1997)* and *t2=(Kate Winslet, Titanic, James Cameron, 1997)*.

Multi-fold Relation Embedding

Multi-fold relation embedding aims to embed the entities, n-fold relations and the instances into continuous low-dimensional vector space (a.k.a. embedding space), and represent these elements as tensors such as vectors or matrices, while the structural information and correlation between them are preserved.

Based on the unified relation representation, we formulate the *multi-fold relation embedding problem in KB* as follows. In order to enhance the expressive ability, we choose three different vector space over field $\mathbb{R}$ (typically real numbers) as the embedding space for entity nodes, fact (or instance) nodes and relations respectively, namely U_e, U_f and U_r. Let $\mathbf{e} \in \mathbb{R}^k$ be an embedding vector of entity node e in U_e, $\mathbf{u} \in \mathbb{R}^l$ be an embedding vector of fact (or instance) node u in U_f, and $\mathbf{r} \in \mathbb{R}^m$ is the embedding vector of relation R_r in U_r.

The objective of multi-fold relation embedding problem is to construct a function $\phi_e : \mathcal{N}_e \to U_e$, a function $\phi_f : \mathcal{N}_f \to U_f$ and a subset $C_r \subset U_r^{\mathcal{M}(R_r)}$ for each relation R_r such that ideally the following properties are satisfied.

1. For every $r \in \mathcal{R}$ and every instance $t \in R_r$, $\phi \circ t \in C_r$, where the symbol $\circ$ denotes function composition.

2. For every $r \in \mathcal{R}$ and every function $t \in \mathcal{N}^{\mathcal{M}(R_r)} \setminus R_r$, $\phi \circ t \notin C_r$.

Here, the function ϕ_e, serving as a representation of $\mathcal{N}_e$, maps an entity e

to its embedding vector $\mathbf{e}$. The function ϕ_f, serving as a representation of $\mathcal{N}_f$, maps a fact node u to its embedding vector $\mathbf{u}$. The subsets $\{C_r : r \in \mathcal{R}\}$, serving as a representation of $\{R_r : r \in \mathcal{R}\}$, define a set of constraints on the embedding vectors which preserve the intra-relational and inter-relational structures of $\{R_r : r \in \mathcal{R}\}$.

Note that each constraint C_r may be identified with a nonnegative cost function $f_r : U^{\mathcal{M}(R_r)} \to \mathbb{R}$ such that

$f_r(\mathbf{t}) = 0$ if $\mathbf{t} \in C_r$, and

$f_r(\mathbf{t}) > 0$ if $\mathbf{t} \notin C_r$

Denote $\Theta := \{f_r : r \in \mathcal{R}\}$. The problem then translates to determing (Θ, ϕ_e, ϕ_f). But $\{R_r : r \in \mathcal{R}\}$ is unknown, and all we have is the observed instances $\{\mathcal{T}_r : r \in \mathcal{R}\}$ and possibly some "negative examples" $\{\mathcal{T}_r^- : r \in \mathcal{R}\}$, where each $\mathcal{T}_r^- \subset \mathcal{N}^{\mathcal{M}(R_r)} \setminus R_r$. Note that when the KB is large, for any $t \in \mathcal{T}_r$, if we replace its value $t(\rho)$ for some role $\rho \in \mathcal{M}(R_r)$ with a random entity, the resulting function falls in $\mathcal{N}^{\mathcal{M}(R_r)} \setminus R_r$ with high probability. This can be used to construct $\mathcal{T}_r^-$.

Treating the problem as learning (Θ, ϕ_e, ϕ_f), we may not need the property 1 above to hold strictly. Then the equality "=0" in (1) is taken as "as close to 0 as possible". Towards a margin-based optimization formulation (which gives better discriminative power and robustness), the threshold 0 in (2) is raised to a positive value c. The problem can then be formulated as finding (Θ, ϕ_e, ϕ_f) to minimize the following global cost function.

$F(\Theta, \phi_e, \phi_f) := \sum_{r \in \mathcal{R}} (\sum_{t \in \mathcal{T}_r} f_r(\phi_e \circ t, \phi_f \circ t) + \sum_{t^- \in \mathcal{T}_r^-} [c - f_r(\phi_e \circ t^-, \phi_f \circ t^-)]_+)$

where $[.]_+$ denotes the rectifier function, namely, $[a]_+ := max(0, a)$.

What remains is to choose a proper space of Θ for this optimization problem, which is at the heart of modeling.

4. Model Description

Towards the goal of multi-fold relation embedding problem, we need to learn three functions, i.e. (Θ, ϕ_e, ϕ_f). The function ϕ_e maps $\mathcal{N}_e$ to U_e, which can be any of the models in "Trans series" such as TransH or DistMult. The function ϕ_f maps $\mathcal{N}_f$ to U_f, and we argue that the fact nodes are correlated with the entities participated, so we use graph convolutional network (GCN) to learn the embedding vector of fact nodes u in $\mathcal{N}_f$. As to function Θ, it models the multi-fold relations and should catch the structural information and correlation between entity nodes, fact nodes and their relations. We model the function as a threefold correlation, namely, to learn the correlation between each two of entity nodes, fact nodes and relation types.

In this section, we first propose a *correlation-constrained embedding framework* for multi-fold relation embedding problem and show how to model the threefold correlations (i.e. correlation between entity nodes and relations, correlation between entity nodes and fact nodes, correlation between fact nodes and relations) in detail. Then, we analyze the complexity of the proposed framework and compare with some of the canonical models. We discuss how to learn the proposed model and an algorithm of high-efficiency is introduced in the end.

4.1. Our Framework

As depicted in Figure 3, based on the *unified relation representation* method, we introduce a *correlation-constrained embedding framework* to embed entity nodes, fact nodes and relation types into three different vector spaces respectively and model their correlations as a whole.

First, we model correlation between entity nodes and relations which reflects the meaning of relation types. Second, we learn the representation of fact nodes by modeling correlation between entity nodes and fact nodes via a GCN model. Finally, correlation between fact nodes and relations are used to model the closeness of fact embeddings mapping to their belonging relation types.

We define the cost function for each instance t as

$$f_r(t) = w^{ER} * g_r^{ER}(t) + w^{EF} * g_r^{EF}(t) + w^{FR} * g_r^{FR}(t), t \in N^{M(R_r)} \tag{7}$$

where $g_r^{ER}(t)$ denotes the loss for modeling correlation between entity nodes and relations, $g_r^{EF}(t)$ denotes the loss for modeling correlation between entity nodes and fact nodes, $g_r^{FR}(t)$ denotes the loss for modeling correlation between fact nodes and their relations.

4.2. Modeling Correlation between Entities and Relations

For an instance $t = (x_1, x_2, ..., x_{|\mathcal{M}(R_r)|}) \in \mathcal{T}_r$, entity node $x_i, i \in |\mathcal{M}(R_r)|$ correlates with relation R_r via instantiating the corresponding role $\rho_i \in \mathcal{M}(R_r)$. Motivated by *m-TransH*, *ProjE* and *HolE* models, we define the following cost function for learning the correlation between entity nodes and relations in a multi-fold relational knowledge base. In detail, we utilize a diagonal combination operator to improve expressivity of the linear combination operator and choose a diagonal matrix $D_r(\rho)$ instead of a real number $a_r(\rho)$ as weight for the projection of entity to each role ρ, i.e. $\mathbb{P}_{n_r}(t(\rho))$.

The cost function $g_r^{ER}(t)$ for each instance t is then defined as

$$g_r^{ER}(t) = \left\| \sum_{\rho \in \mathcal{M}(R_r)} D_r(\rho) \mathbb{P}_{n_r}(t(\rho)) + b_r \right\|_2^2, t \in \mathcal{N}_e^{\mathcal{M}(R_r)} \tag{8}$$

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

$$\mathbb{P}_{n_r}(z) = z - n_r^\top z n_r \tag{9}$$

n_r and b_r are unit length orthogonal vectors in U, $D_r \in \mathbb{R}^{k \times k}$ is a function that

$$\sum_{\rho \in \mathcal{M}(R_r)} D_r(\rho) = \mathbf{0} \tag{10}$$

Then the objective can be defined as a margin-based optimization problem:

$$\mathcal{L}_{ER} = \sum_{r \in R} (\sum_{t \in \mathcal{T}_r} g_r^{ER}(t) + \sum_{t^- \in \mathcal{T}_r^-} [c - g_r^{ER}(t^-)]_+) + \frac{\lambda}{2} \sum_{r \in R} (\| diag(\sum_{\rho \in \mathcal{M}(R_r)} D_r(\rho)) \|^2) \tag{11}$$

where $[x]_+ = max(0, x)$, function $diag(X)$ get the leading diagonal vector, $\lambda \in [0, 1]$ is a balance factor.

4.3. Modeling Correlation between Entities and Facts

Entity node $x_i \in N_e, i \in |\mathcal{M}(R_r)|$ links to fact/instance node $t \in N_f$ via roles of relation $r \in R_r$. To model the correlation between entity nodes and their linked fact (or instance) nodes, we utilize the structure characteristics of multi-fold relations in knowledge bases and employ graph convolutional network (GCN) to learn the representation of fact nodes. GCNs [24, 25] are capable of learning the structure of local graph neighborhoods for large-scale relational data and are usually described as a differentiable message-passing framework:

$$h_i^{(l+1)} = \delta(\sum_{m \in \mathcal{M}_i} g_m(h_i^{(l)}, h_j^{(l)})) \tag{12}$$

where $h_i^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of node v_i in the l-th layer of the neural network, with $d^{(l)}$ being the dimensionality of this layer's representations. Incoming messages of the form $g_m(\cdot, \cdot)$ are accumulated on the set of neighbors for node v_i, i.e. $\mathcal{M}_i$, and passed through an element-wise activation function $\delta(\cdot)$, such as ReLU or Sigmoid.

In the circumstance of multi-fold relations in the unified representation, entity nodes' neighborhoods are consist of different nodes of instances they participated in with role $\rho \in \mathcal{M}(R_r)$ and vice versa for the instance nodes. Thus we extend R-GCN framework [26] to modeling multi-fold relation and propose the following model for calculating the forward-pass update of an entity node (or instance node) denoted by v_i in a multi-fold relational knowledge base:

$$h_i^{(l+1)} = \delta(\sum_{r \in R} \sum_{\rho \in \mathcal{M}(R_r)} \sum_{j \in N_i^\rho} \frac{1}{c_r * c_{i,\rho}} W_\rho^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}) \tag{13}$$

where N_i^{ρ} denotes the set of neighbor indices of the node i under role $\rho \in \mathcal{M}(R_r)$ of relaiton r. c_r and $c_{i,\rho}$ are problem-specific normalization constant chosen here as $c_r = |R|$ and $c_{i,\rho} = |N_i^{\rho}|$.

Following R-GCN framework, we add a single self-connection of a special relation type to each node in the data to ensure the message passing from layer l to layer $l+1$. To address the parameters explosion problem, we also apply *bisis-decomposition* and *block-diagonal-decompostion* methods [26] for regularizing the weights of each layers.

The model takes one-hot vector for each entity (or instance) node in the graph as input and stack L GCN layers as defined above, we choose DistMult factorization [27] as the output layer and define the cost function of each instance t of n-ary relation ($n \geq 2$) r as an average cost of triples for each role $\rho \in \mathcal{M}(R_r)$:

$$g_r^{EF}(t) = \frac{1}{|\mathcal{M}(R_r)|} \sum_{\rho \in \mathcal{M}(R_r)} f(e_{t,\rho}, \rho, e_{t,f}) \tag{14}$$

where $f(e_{t,\rho}, \rho, e_{t,f}) = \mathbf{e}_{t,\rho}^T M_r(\rho) \mathbf{e}_{t,f}$ with $\mathbf{e}_{t,\rho}$ denotes embedding vector of entity $e_{t,\rho}$ in role ρ and $\mathbf{e}_{t,f}$ denotes fact embedding of the instance t. Each role ρ is associated with a diagonal matrix $M_r \in \mathbb{R}^{d \times d}$.

We sample w negative examples for each observed (positive) instance by randomly corrupting either entity of each positive example and utilize cross-entropy loss bellow as our optimization objective to enforce the model to score positive examples higher than the negative ones:

$$\mathcal{L}_{EF} = -\frac{1}{(1+w)|\mathcal{T}|} \sum_{r \in R} (\sum_{t \in \mathcal{T}_r} logl(g_r^{EF}(t)) + \sum_{t^- \in \mathcal{T}_r^-} log(c - l(g_r^{EF}(t^-)))) \tag{15}$$

where $\mathcal{T}$ is the total set of observed examples and consist of instances in $\mathcal{T}_r$ for every $r \in R$, l is the logistic sigmoid function, c is a constant chosen as margin such as 1.0.

4.4. Modeling Correlation between Facts and Relations

We argue that facts should be embedded close to their relations and fact embeddings of the same relation should aggregate in a cluster, otherwise far away from each other. To measure the similarity between embedded facts and relations, we employ *dot* product to calculate the distance of fact embeddings $\mathbf{e}_{t,f}$ and relation embeddings $\mathbf{r}_t$ of instance t. We define the cost function as

$$g_r^{FR}(t) = \mathbf{e}_{t,f}^T \mathbf{r}_t \tag{16}$$

Then we define the optimization objective for a single instance t as a margin-based problem:

$$\mathcal{L}_t = max\{0, c - [g_r^{FR}(t) - g_{r^-}^{FR}(t^-)]\} \tag{17}$$

where t^- is a negative instance in which we replace r in t by an n-ary relation r^- $(r^- \neq r)$.

We also assume that diagonal matrix $D_r(\rho)$ to be similar to diagonal matrix $M_r(\rho)$ and define the distance by absolute difference of their normalized trace (the trace of matrix characterizes its similarity invariance **addreferencehere**):

$$d(M_r(\rho), D_r(\rho)) = \|tr(M_r(\rho) - tr(D_r(\rho)))\| \tag{18}$$

where function $tr(\cdot)$ denotes the normalized trace of a matrix. The optimization objective for each relation r is then defined as

$$\mathcal{L}_r(\rho) = max\{0, c - [d(M_r(\rho), D_r(\rho)), d(M_r(\rho), D_{r^-}(\rho^-))]\} \tag{19}$$

where $r^- \in R$ is an n-ary relaton different from r, $\rho^- \in \mathcal{M}(R_{r^-})$.

We then define the total loss function as

$$\mathcal{L}_{FR} = \sum_{r,r^- \in R, r \neq r^-} (\sum_{t \in \mathcal{T}_r} \mathcal{L}_t + \sum_{\rho \in \mathcal{M}(R_r)} \mathcal{L}_r(\rho)) \tag{20}$$

4.5. Joint Optimization Problem

Our goal is to embed all the entities $e \in N_e$, facts $u \in N_f$ and multi-fold relations $r \in N_r$ in knowledge base $\mathcal{G}$ into d-dimensional entity vector space U_e, fact vector space U_f and relation vector space U_r. As the embedding vectors of entities, facts and relations are shared across the proposed framework, our solution is to collectively minimize the three optimization objectives $\mathcal{L}_{ER}$,

$\mathcal{L}_{EF}$, $\mathcal{L}_{FR}$. To achieve the goal, we formulate a joint optimization problem as minimizing the weighted combination of the three objectives:

$$min\, \mathcal{L} = \lambda_{ER} * \mathcal{L}_{ER} + \lambda_{EF} * \mathcal{L}_{EF} + \lambda_{FR} * \mathcal{L}_{FR} \quad (21)$$

The learning of entity, fact and relation embeddings can be mutually influenced via joint optimizing the global objective $\mathcal{L}$, which reduces the errors in each component and promotes more powerful representations.

5. Model Learning and Complexity Analysis

5.1. Learning Method Discussion

To solve the joint optimization problem in Eq.(21), an intuitive solution is to minimize the three objectives sequentially, i.e. first learn the correlations of entities and multi-fold relations via optimizing $\mathcal{L}_{ER}$ on each instances, then utilize the learned embedding of entities to minimize $\mathcal{L}_{EF}$ and train a GCN model on the whole knowledge base, finally, apply the acquired representations of entities, facts and relations to optimize $\mathcal{L}_{FR}$. However, such a solution can hardly converge (training process on $\mathcal{L}_{FR}$ will update the embeddings of facts and relations which may destroy the convergence of $\mathcal{L}_{ER}$ and $\mathcal{L}_{EF}$). Moreover, the learning procedure does not fully exploit the correlation between facts and relations expressed in $\mathcal{L}_{FR}$ to provide mutual feedbacks when minimizing $\mathcal{L}_{ER}$ and $\mathcal{L}_{EF}$.

Thus we follow CoType [28] and exploit a stochastic sub-gradient descent algorithm based on edge sampling strategy to efficiently solve Eq.(21), which can be proved to converge to the local minimum. In detail, we iteratively sample from each of the three objectives $\mathcal{L}_{ER}, \mathcal{L}_{EF}, \mathcal{L}_{FR}$ a batch of positive instances (e.g., (e_i, f_j, r_k)) and generate V negative samples for each positive ones based on *the Closed World Assumption*, i.e. replace any one of the entities(or fact node) involved in a multi-fold relation instance with other entities or fact nodes to get a new instance that is not exist in the training set. We then update each embedding vector based on the derivatives. The model learning process of **CoRelatE** is summarized in Algorithm 1.

5.2. Computational Complexity Analysis

The objective function should be regarded as an optimization task and be solved by proposing a new algorithm, which employs the proposed tricks to learn the above three correlations simultaneously. The computation complexity can be discussed at meanwhile.

Converting multi-fold relations to binary relations results in a heterogeneity of the predicates, unfavorable for knowledge base embedding. M-TransH [11] 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 $Eq.(7)$:

$$\begin{aligned} f_r\left(t\right) &= \left\| \sum_{\rho \in M(R_r)} a_r \mathbb{P}_{n_r}(t(\rho)) + b_r \right\|_2^2 + \beta * g_r(t), \\ t &\in N^{M(R_r)} \end{aligned} \tag{22}$$

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

As to the constraint 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, it requires the Euclidean distance between the embedded fact vector $\mathbb{P}_{n_r}(e_{fact})$ and its corresponding relation vector r to be zero. Namely,

$$g_r(t) = \left\| r - \mathbb{P}_{n_r}(e_{fact}) \right\|_2, t \in N^{M(R_r)} \tag{23}$$

- **Radius Constraint**

Radius constraint adopts a relaxed constraint on the Euclidean distance between $\mathbb{P}_{n_r}(e_{fact})$ and r. If the fact is an positive instance of the relation r, we need the distance to be smaller than ϵ, otherwise much bigger than ϵ. In this way, we define $g_r(t)$ as $Eq.(9)$,

$$g_r(t) = \max(0, \|r - \mathbb{P}_{n_r}(e_{fact})\|_2 - \epsilon), t \in N^{M(R_r)} \tag{24}$$

• **Cosine Constraint**

Cosine constraint exploits the cosine distance as measurement, it renders the distance of the embedded fact vector $\mathbb{P}_{n_r}(e_{fact})$ and its corresponding relation vector r to be near zero. Namely,

$$g_r(t) = \cos\langle r, \mathbb{P}_{n_r}(e_{fact})\rangle, t \in N^{M(R_r)} \tag{25}$$

We present an illustration of Group-constrained Embedding methods in Figure 1, which consists of 4 subgraphs, i.e. graph A, B, C and D. The first graph A shows the structure of the entities and multi-fold relations in the origin vector space. The other three graphs show the Group-constrained Embedding of multi-fold relations with Zero Constraint, Radius Constraint or Cosine Constraint methods respectively.

In the origin 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 the origin vector space. 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 origin 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 hyper sphere, "$relation1$" 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 the same, falling onto a straight line when projected to a two-dimensional plane.

5.3. Proposed Model

Using the Group-constrained Embedding method, we 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 circle, the relation vector act as the center of the circle, 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 smaller than ϵ on the hyperplane, otherwise much bigger than ϵ. The cost function f_r is defined as $Eq.(11)$.

$$\begin{aligned} f_r(t) = &\left\| \sum_{\rho \in M(R_r)} a_r \mathbb{P}_{n_r}(t(\rho)) + b_r \right\|_2^2 + \\ &\beta * \max(0, \|b_r - \mathbb{P}_{n_r}(e_{fact})\|_2 - \epsilon), \\ &t \in N^{M(R_r)} \end{aligned} \tag{26}$$

Where $\mathbb{P}_{n_r}$ is defined by TransH, namely $Eq.(5)$. Obviously, $Eq.(11)$ is converted from $Eq.(7)$ by setting the constraint term $g_r(t)$ to $Eq.(9)$.

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 constraint term $g_r(t)$. Namely, with Zero Constraint method, the model Gm-TransH sets $g_r(t)$ to $Eq.(8)$, the cost function f_r is defined as $Eq.(12)$

$$\begin{aligned} f_r\left(t\right) = &\left\| \sum_{\rho \in M(R_r)} a_r \mathbb{P}_{n_r}(t(\rho)) + b_r \right\|_2^2 + \\ &\beta * \left\| b_r - \mathbb{P}_{n_r}(e_{fact}) \right\|_2 , \\ &t \in N^{M(R_r)} \end{aligned} \tag{27}$$

We call the Group-constrained m-TransH model with Zero Constraint **Gm-TransH:zero**.

Similarly, with the Cosine Constraint method, the model Gm-TransH sets $g_r(t)$ to $Eq.(10)$, the cost function f_r is defined as $Eq.(13)$

$$\begin{aligned} f_r\left(t\right) = &\left\| \sum_{\rho \in M(R_r)} a_r \mathbb{P}_{n_r}(t(\rho)) + b_r \right\|_2^2 + \\ &\beta * \cos \left\langle r, \mathbb{P}_{n_r}(e_{fact}) \right\rangle , \\ &t \in N^{M(R_r)} \end{aligned} \tag{28}$$

We call the Group-constrained m-TransH model with Cosine Constraint **Gm-TransH:cosine**.

5.4. 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 [11] for each instance with multi-fold relation, resulting in several triples with binary relations, which are appropriate for these models. After a S2C conversion,

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 triples 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 [16]	$O(N_e m + N_{r2}(2k + 2nk))$	$O((2mk + k)N_{t2})$
NTN [16]	$O(N_e m + N_{r2}(n^2 s + 2ns + 2s))$	$O(((m^2 + m)s + 2mk + k)N_{t2})$
TransE [6]	$O(N_e m + N_{r2} n)$	$O(N_{t2})$
TransH [7]	$O(N_e m + 2N_{r2} n)$	$O(2mN_{t2})$
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 [20]	$O(2N_e m + 2N_{r2} n)$	$O(2nN_{t2})$
m-TransH [11]	$O((N_e + N_f)m + 2N_r n + N_\rho)$	$O(mN_\rho)$
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	$O((N_e + N_f)m + 2N_r n + N_\rho)$	$O(m(N_\rho + 3N_t))$

the number of instances/triples and relations are changed as follows:

$$N_{r2} = \sum_{i=1}^{N_r} \frac{n_{ri} * (n_{ri} - 1)}{2}, ri \in R \tag{29}$$

$$N_{t2} = \bigcup_{i=1}^{N_t} \frac{n_{ti} * (n_{ti} - 1)}{2}, ti \in N^{M(R_r)} \tag{30}$$

$$N_\rho = \sum_{i=1}^{N_r} n_{ri}, ri \in R \tag{31}$$

Where n_{ri} denotes the fold of the i-th relation ri, n_{ti} denotes the fold of the i-th instance ti, $N_r \ll N_{r2}$ and $N_t \ll N_{t2}$.

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 35,35 and 42 minutes respectively, which are close to transH and m-TransH(30 minutes) models, but outperform the existing methods on link prediction and relation classification tasks significantly.

6. Experiments and Analysis

In this section, we empirically study and evaluate our approach on two tasks: link prediction and instance classification.

Table 2: Statistics of the extended JF17K dataset.

Dataset	$\mathbf{G}^{\checkmark}_{\mathbf{s2c}}/\mathbf{G}^{?}_{\mathbf{s2c}}$	$\mathbf{G}^{\checkmark}/\mathbf{G}^{?}$	$\mathbf{G}^{\checkmark}_{\mathbf{id}}/\mathbf{G}^{?}_{\mathbf{id}}$	$\mathbf{G}^{\checkmark}_{\mathbf{fact}}/\mathbf{G}^{?}_{\mathbf{fact}}$
# 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 origin 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

6.1. Datasets

JF17K. We use a cleaned and extended JF17K datasets [11] in our experiments. The original JF17K datasets were transformed from the full RDF formatted Freebase data. Denote the fact representation by F. Two instance representations $T(F)$ (denoted by G), $T_{id}(F)$ (denoted by G_{id}) and a triple representation $S2C(G)$ (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 contain many redundant samples, which may affect the results, we first cleaned up the repetitive data. 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 a 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 instances. Two instances which share the same relation and the same entities except one role were assigned a same fact node. We call the extended set G_{fact} and divide it into training set $G^{\checkmark}_{fact}$ and testing set $G^{?}_{fact}$. The statistics of these datasets are shown in Table 2.

FB15K. We also use FB15K dataset [6] on instance classification task. Since FB15K dataset contains only triples with binary relations and has no fact nodes in the triples, we extend the FB15K dataset by adding an unique fact node to each triple. Thus, we can use the extended FB15K to train the proposed Gm-TransH model and test its performance. We use the origin FB15K dataset to train the NTN, TransE, TransH and TransR models, for convenience, we use "*Raw*" to denote the origin FB15K dataset and use "*Ext*" to denote the extended FB15K dataset. Table 3 lists the statistics of the origin and extended FB15K datasets.

6.2. Link Prediction

Link prediction aims to complete the missing entities for instances or triples, i.e., predict one entity given other entities and the relation. For example, for

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

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

triple (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.

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 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 using the proposed models and then sorted by ascending order. Then we use Hit@10(HIT) and Mean Rank (RANK) [6] of the correct entities ranked as the performance metrics to evaluate the proposed models. The two metrics are commonly used to evaluate the performance of knowledge base embeddings. Hit@10 computes the probability of the positive entities that rank the top 10% for all the entities. Mean Rank means the average position of the positive entities ranked.

Implementation. We conduct eight kinds of experiments in this task, the training and testing datasets for each of the experiments 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, the batch size B among 120, 480, 960, 1920. The optimal configurations of the three 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 Figure 2 and 3, which show 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 multi-fold relation embeddings. Furthermore, Gm-TransH:zero obtains better performance than Gm-TransH:radius and Gm-TransH:cosine in most cases, showing that Zero Constraint outperforms Radius Constraint and Cosine Constraint.

6.3. 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 [16, 7]

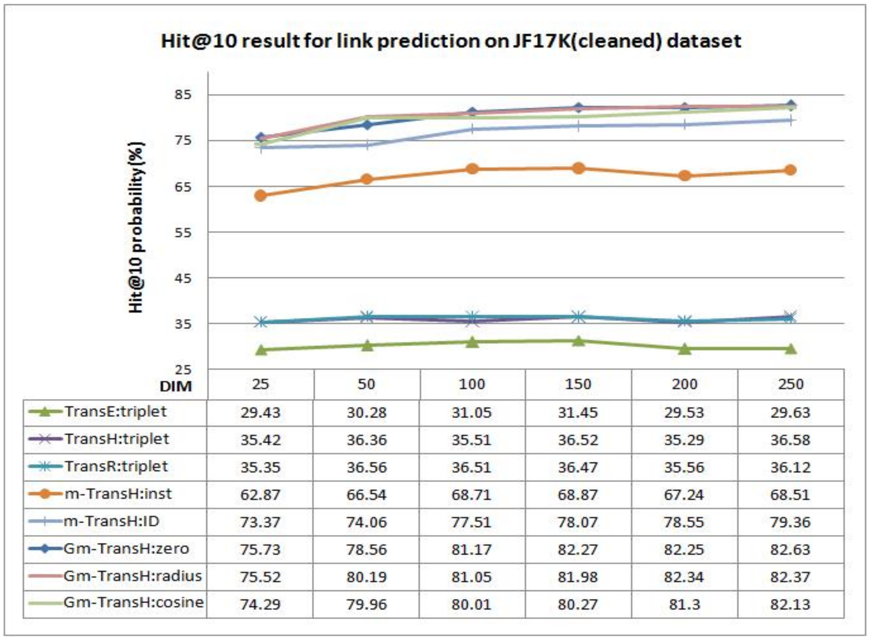

Figure 3: The probability of ranking the top 10% for different embedding dimensions.

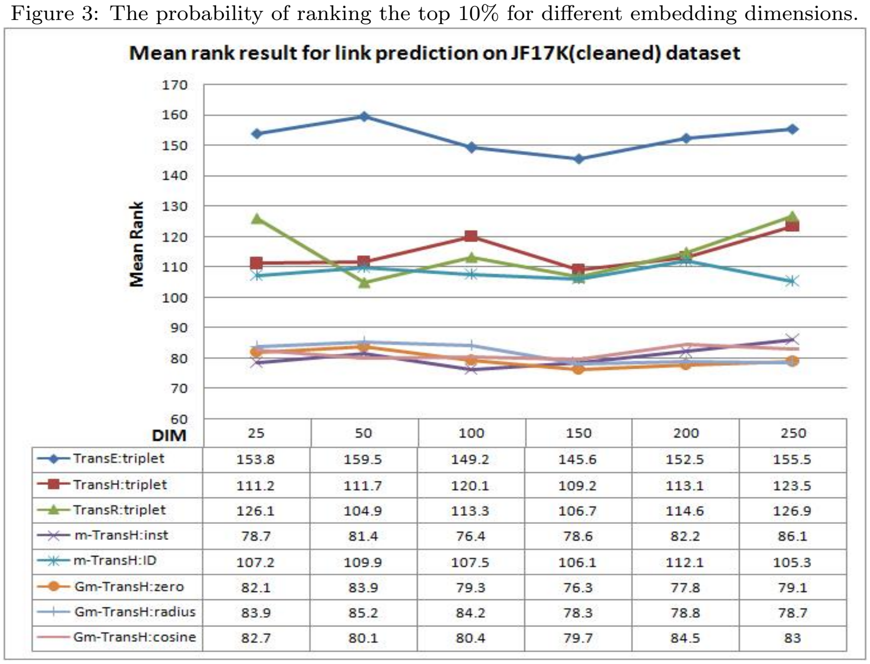

Figure 4: The mean rank for different embedding dimensions.

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 contain only positive instances, we construct negative instances following the same procedure used for FB13 in [16]. 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 triples, we train and evaluate the NTN, Trans(E, H, R) models on the origin FB15K dataset (denoted as *Raw*) and the G_{s2c} dataset of JF17K. We use the NTN code released by Socher [16] 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 [11] 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 5 lists the evaluation results of instance classification in detail. We can observe that on both FB15K and JF17K datasets, the Gm-TransH models outperforms the baseline models including NTN, Trans(E, H, R) and m-TransH significantly. The accuracy can reach more than 90% on Gm-TransH:zero and Gm-TransH:cosine, achieving a new state-of-the-art performance. Moreover, from the results on the FB15K(*Raw*) and the FB15K(*Ext*) datasets, we see that even for binary relations (i.e. multi-fold relations whose fold equals 2), the Group-constrained Embedding method is practicable and reliable.

Table 5: 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	89.2
Gm-TransH:radius	—	89.3	88.9
Gm-TransH:cosine	—	90.1	91.3

7. Conclusions and Future Work

We presented a Group-constrained Embedding method for multi-fold relations and proposed a new representation learning framework Gm-TransH using the optimizing method. We evaluate the effectiveness and performance of the proposed methods and models on extended FB15K and JF17K datasets. Experimental results show that the Gm-TransH models outperforms all baseline models on link prediction task 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.

Acknowledgements

This work was supported by the Science and Technology Program of Shenzhen of China under Grant Nos. JCYJ20170818160208570 and JCYJ20170307160458368.

References

[1] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, international conference on management of data (2008) 1247–1250.

[2] G. A. Miller, Wordnet: a lexical database for english, Communications of The ACM 38 (11) (1995) 39–41.

[3] F. M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, international world wide web conferences (2007) 697–706.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. G. Ives, Dbpedia: a nucleus for a web of open data, international semantic web conference (2007) 722–735.

[5] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, S. Liu, Modeling relation paths for representation learning of knowledge bases, empirical methods in natural language processing (2015) 705–714.

[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in Neural Information Processing Systems (2013) 2787–2795.

[7] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, national conference on artificial intelligence (2014) 1112–1119.

[8] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, national conference on artificial intelligence (2015) 2181–2187.

[9] B. Shi, T. Weninger, Proje: Embedding projection for knowledge graph completion, national conference on artificial intelligence (2017) 1236–1242.

[10] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, international conference on machine learning (2016) 2071–2080.

[11] J. Wen, J. Li, Y. Mao, S. Chen, R. Zhang, On the representation and embedding of knowledge bases beyond binary relations, international joint conference on artificial intelligence (2016) 1300–1307.

[12] B. Fatemi, P. Taslakian, D. Vazquez, D. Poole, Knowledge hypergraphs: Extending knowledge graphs beyond binary relations., neural information processing systems.

[13] J. Rouces, G. De Melo, K. Hose, Framebase: Representing n-ary relations using semantic frames, european semantic web conference (2015) 505–521.

[14] A. Bordes, X. Glorot, J. Weston, Y. Bengio, A semantic matching energy function for learning with multi-relational data, neural information processing systems 94 (2) (2014) 233–259.

[15] A. Bordes, J. Weston, R. Collobert, Y. Bengio, Learning structured embeddings of knowledge bases, Proceedings of the Twenty-fifth AAAI Conference on Artificial Intelligence.

[16] R. Socher, D. Chen, C. D. Manning, A. Y. Ng, Reasoning with neural tensor networks for knowledge base completion, neural information processing systems (2013) 926–934.

[17] A. Bordes, X. Glorot, J. Weston, Y. Bengio, Joint learning of words and meaning representations for open-text semantic parsing, international conference on artificial intelligence and statistics (2012) 127–135.

[18] R. Jenatton, N. L. Roux, A. Bordes, G. Obozinski, A latent factor model for highly multi-relational data, neural information processing systems (2012) 3167–3175.

[19] H. Xiao, M. Huang, Y. Hao, X. Zhu, Transa: An adaptive approach for knowledge graph embedding., arXiv: Computation and Language.

[20] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, international joint conference on natural language processing (2015) 687–696.

[21] G. Ji, K. Liu, S. He, J. Zhao, Knowledge graph completion with adaptive sparse transfer matrix, national conference on artificial intelligence (2016) 985–991.

[22] S. He, K. Liu, G. Ji, J. Zhao, Learning to represent knowledge graphs with gaussian embedding, conference on information and knowledge management (2015) 623–632.

[23] Q. Xie, X. Ma, Z. Dai, E. H. Hovy, An interpretable knowledge transfer model for knowledge base completion, meeting of the association for computational linguistics 1 (2017) 950–962.

[24] D. Duvenaud, D. Maclaurin, J. Aguileraiparraguirre, R. Gomezbombarelli, T. D. Hirzel, A. Aspuruguzik, R. P. Adams, Convolutional networks on graphs for learning molecular fingerprints, neural information processing systems (2015) 2224–2232.

[25] T. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, international conference on learning representations.

[26] M. Schlichtkrull, T. Kiph, P. Bloem, R. v. d. Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, European Semantic Web Conference.

[27] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, international conference on learning representations.

[28] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss, H. Ji, T. F. Abdelzaher, J. Han, Cotype: Joint extraction of typed entities and relations with knowledge bases, the web conference (2017) 1015–1024.