Knowledge Graph Embedding by Double Limit Scoring Loss

Xiaofei Zhou, Lingfeng Niu, Qiannan Zhu, Xingquan Zhu, Ping Liu, Jianlong Tan, and Li Guo

Abstract—Knowledge graph embedding is an effective way to represent knowledge graph, which greatly enhance the performances on knowledge graph completion tasks, e.g., entity or relation prediction. For knowledge graph embedding models, designing a powerful loss framework is crucial to the discrimination between correct and incorrect triplets. Margin-based ranking loss is a commonly used negative sampling framework to make a suitable margin between the scores of positive and negative triples. However, this loss can not ensure ideal low scores for the positive triplets and high scores for the negative triplets, which is not beneficial for knowledge completion tasks. In this paper, we present a double limit scoring loss to separately set upper bound for correct triplets and lower bound for incorrect triplets, which provides more effective and flexible optimization for knowledge graph embedding. Upon the presented loss framework, we present several knowledge graph embedding models including TransE-SS, TransH-SS, TransD-SS, ProjE-SS and ComplEx-SS. The experimental results on link prediction and triplet classification show that our proposed models have the significant improvement compared to state-of-the-art baselines.

Index Terms—Knowledge graph, embedding, representation learning, knowledge graph completion, loss function

1 Introduction

Knowledge graphs (KGs) related researches have become one of the major issues in artificial intelligence, due to its effectiveness in representing, learning and predicting structured information as well as building knowledge database. In recent years, various available large-scale KGs such as NELL [1], ProBase [2], [3], GeneOntology [4] and Yago [5] have become very important resources to support many AI related applications, such as question answering system [6], [7], [8], [9], WebSearch [10], [11] and Information Extraction [12], [13], [14], [15], [16].

KGs are composed of entities and relations, where entities are nodes and relations are edges. A triplet (h, r, t) is a basic structural unit of KGs that represents a relationship r from head entity h to tail entity t. Looking for more effective representations of KGs, which are different from the symbolic manner (h, r, t), are particularly important for knowledge graph completion (KGC) tasks, such as link prediction and triplet classification.

KG embedding is a promising approach to model the relational facts into continuous vector space [17], [18], [19], [20], [21], [22], [23]. In a KG embedding model, there are two major components, the scoring triplets and the optimizing loss function. In the last few years, negative sampling framework is very common for modeling KG embedding. In this framework, for a correct triplet (h, r, t), a corresponding incorrect triplet (h', r, t') can be sampled from the data set, then the pair of triplets $\{(h, r, t), (h', r, t')\}$ as input are measured by the scoring function and optimized by the loss function. Fig. 1 simply demonstrates the process.

- The scoring function for a triplet is denoted as $f_r(h, t)$ which measures the matching of a triplet (h, r, t) in the embedding space. The score is usually low when a triplet (h, r, t) is correct, otherwise high for the corrupted triplet (h', r, t'). The ideal score of a positive triplet (h, r, t) is usually lower than its corresponding negative triplet (h', r, t').
- The loss function for optimization is to distinguish the scores of positive and negative triplets. One can learn the embeddings of entities and relations by minimizing a designed loss, such as margin-based ranking loss function $max(0, \mu + f_r(h, t) - f_r(h', t'))$ which expects that the score of positive triplet (h, r, t) is lower at least by μ than that of corresponding negative triplet.

Most of current negative sampling KG embedding models focus on the first component that is to design effective scoring functions $f_r(h, t)$ for triplets, such as structured embedding (SE) [24], latent factor model (LFM) [25], [26], semantic matching model (SME) [27], [28], neural tensor network model (NTN) [17] and translation models [18], [19], [20], [21] etc. There are fewer researches on loss functions, and the margin-based ranking loss is the most common negative sampling method for KG embedding models.

The margin-based ranking loss (abbreviated as L_R loss in this paper) has achieved great success in KG embedding learning, but this loss maybe lead to the overlapping

- Xiaofei Zhou, Qiannan Zhu, Ping Liu, Jianlong Tan, and Li Guo are with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: {zhouxiaofei, zhuqiannan, liuping, tanjianlong, guoli}@iie.ac.cn.
- Lingfeng Niu is with the Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China. E-mail: nlf@ucas.ac.cn.
- Xingquan Zhu is with the Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA. E-mail: xzhu3@fau.edu.

Manuscript received 22 Aug. 2019; revised 8 Jan. 2021; accepted 9 Feb. 2021. Date of publication 22 Feb. 2021; date of current version 7 Nov. 2022.
(Corresponding author: Xingquan Zhu.)
Recommended for acceptance by A. Poulovassilis.
Digital Object Identifier no. 10.1109/TKDE.2021.3060755

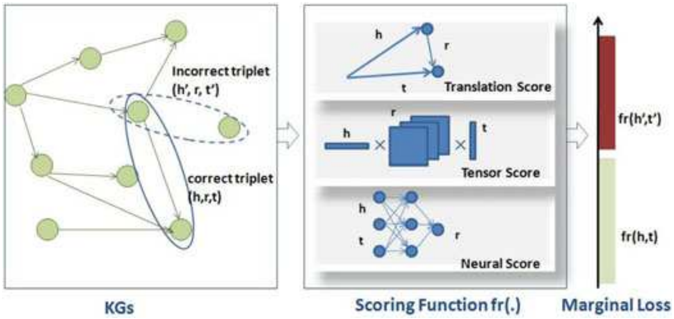


Fig. 1. Example of knowledge graph embedding method.

between the positive and negative scores, and especially cannot ensure that the positive triplet's score $f_r(h,t)$ is low enough for a golden triplet (h,r,t) [29]. Under a margin-based ranking loss, there maybe three kinds of value distributions for a pair of positive and negative triplets $\{f_r(h,t), f_r(h',t')\}$, including $\{lowf_r(h,t), highf_r(h',t')\}$, $\{lowf_r(h,t), lowf_r(h',t')\}$ and $\{highf_r(h,t), highf_r(h',t')\}$. Among the three cases, $\{lowf_r(h,t), lowf_r(h',t')\}, \{highf_r(h,t), highf_r(h',t')\}$ are both unexpected for entity prediction, because low $f_r(h',t')$ and high $f_r(h,t)$ are very prone to the overlapping of the scores of the positive and negative entities. TransE-RS and TransH-RS [29] presented to add a upper-limit scoring loss on $f_r(h,t)$ to guarantees low scores for the golden triplets, which can effectively avoid $\{highf_r(h,t), highf_r(h',t')\}$ case, and has significant improvement compared to state-of-the-art baselines. But this combined loss (abbreviated as L_{RS} loss) still can not avoid the $\{lowf_r(h,t), lowf_r(h',t')\}$ case. Inspired by the upper-limit scoring of L_{RS} on the positive triplets, this paper also sets a lower-limit score for negative triplets to further avoid $\{lowf_r(h,t), lowf_r(h',t')\}$ case, which can rank the correct entities before the incorrect, and also provide more flexible scoring control for the triplets. The intuitive examples and detailed explanations are give in Section 3.4.

In this paper we present a novel double limit scoring loss L_{SS} for KG embedding learning, which has an upper-limit score for positive triplets and a lower-limit score for negative triplets. Upon our L_{SS} loss, we present our models TransE-SS, TransH-SS, TransD-SS, ProjE-SS and ComplEx-SS, by extending TransE [18], TransH [19], TransD [19], ProjE [30] and ComplEx [31]. The effectiveness of the extended models is evaluated on link prediction and triple classification, using standard benchmark datasets of WordNet [32] and Freebase [33]. The main contributions of this paper can be summarized as follows.

- A double limit scoring loss framework for KG embedding models is presented, which enables positive and negative triplets to be more flexibly and effectively optimized.
- Based on the double limit scoring loss, we derive several new extended models TransE-SS, TransH-SS, TransD-SS, ProjE-SS and ComplEx-SS from the current state-of-the-art models. Our models improve the performances of the algorithms without increasing the parameter and computation complexity.
- Experiments are carried out on the datasets WordNet and Freebase, and the results show that our proposal models have significant improvements comparing with previous baselines in KG completion tasks.

The rest of this paper is organized as follows. In the next section, some related works about KG embedding models are given. The proposed L_{SS} loss is presented and discussed in Section 3. Upon L_{SS} loss framework, some new models including TransE-SS, TransH-SS, TransD-SS, ProjE-SS and ComplEx-SS for KG embedding are also presented. In Section 4, we detail the experimental studies on our models and make discussions on how parameters affect the performance of our model. Finally we give a conclusion for our paper in the last section.

2 Related Works

2.1 Knowledge Graph Embedding Models

According to the definitions of different scoring functions, the KG embedding models can be briefly classified into translation models and semantic matching models.

The translation models define a score function to measure the distance between two entities, and usually each relation is regarded as a translation operator. TransE [18] is the most widely used translational model that regards a relationship as a translation operation for a head-tail entity pair. This model is very efficient to 1-to-1 relation, but it has issues for N-to-1, 1-to-N and N-to-N relations [19]. To address the issues of TransE for complex relations, TransH [19] introduces relation-specific hyperplanes with normal vector to enable an entity to have different roles. Furthermore, TransR/CTransR [34] considers the relation-specific spaces instead of the hyperplane in TransH. TransD [20] replaces transfer matrices by the product of two projection vectors for an entity-relation pair. TranSparse [21] simplifies TransR by adopting sparse transfer matrices to project the head and tail entities. STransE [35], utilizes transfer matrices to map the head and tail embeddings into relation spaces separately. ITransF [36] is a generalization of STransE, which introduces sparse attention vectors for each relation. TransF [22] weakens the scoring for the translated head and the inverse translated tail.

The semantic matching models define a score function as similarity by matching latent semantics of entities and relations. RESCAL [37] is a tensor factorization model that encodes each relation as a matrix to model pairwise interactions between latent semantics of entities. Some other tensor factorization based works have been proposed [38], [39], [40], [41], [42], [43], [44], [45], [46]. Semantic Matching Energy (SME) [27], [28] aims to capture correlations between entities and relations via multiple matrix products and Hadamard product, which considers both linear and bilinear form of semantic matching energy functions for optimization. Single Layer Model (SLM) [17] introduces single layer of neural network for semantic matching. NTN Model (NTN) [17] extends the SLM by considering the second-order correlations into nonlinear transformation. However, the complexity of the model is much higher, making it difficult to handle large scale graphs. DistMult [47] considers second-order correlations between entity embeddings by using a quadratic form, and defines a bilinear matching function as scoring function. HolE [48] takes full advantages of the powerful expression of RESCAL and the simplicity of DistMult to embed entities and relations as vectors. ComplEx [31] is an extension of DistMult, which embeds entities and relations into a complex-valued space for better modeling asymmetric relations. ProjE [30] is the state-of-the-art neural based model for learning KG embeddings, which is a neural network with a combination layer and a projection (i.e., output) layer. ConvE [49] uses convolutional and fully-connected layers to model the interactions between input entities and relationships. Multiple-step relation path based models such as PTransE [50], RTransE [51] and Goal-Directed Random Walk [52] can achieve excellent performance with extended path information, but the size of these models grow exponentially as the path-length increases. Recently RotatE [53] presents a Hadmard product scoring function to model and infer all the various relation patterns of KG, including symmetry, inversion and composition, which significantly outperform existing state-of-the-art models for link prediction.

2.2 Loss Functions

From the view of negative sampling optimization for KG embedding, we summarize the related losses as follows.

Margin-based ranking loss L_R is a usual used loss function for KG embedding models, which has successfully been used for LFM [26], SME [27], NTN [17], TransE [18], TransH [19], TransR [34], and TransD [20] etc. The L_R loss is :

$$L_R = \sum_{\substack{(h,r,t)\in\Delta \\ (h',r,t')\in\Delta'}} [\mu_1 + f_r(h,t) - f_r(h',t'))]_+, \quad (1)$$

where $[x]_+ = max(0,x)$ is a rectified linear unit that denotes the positive part of x. $\mu_1 > 0$ is the margin between the positive and negative triplets, and Δ is the set of positive triplets. $\Delta' = \{(h',r,t)|h' \in E\} \cup \{(h,r,t')|t' \in E\}$ denotes the set of corrupted triplets, including training triplets with either the head or tail replaced by a random entity.

TransE-RS and TransH-RS [29] present to add a limit-based scoring loss term $[f_r(h,t) - \mu_2]_+$ for positive triplet into the margin-based ranking loss framework. The combined loss framework (abbreviated as L_{RS}) for KG embedding is:

$$L_{RS} = \sum_{\substack{(h,r,t)\in\Delta; \\ (h',r,t')\in\Delta'}} [\mu_1 + f_r(h,t) - f_r(h',t')]_+ + \lambda[f_r(h,t) - \mu_2]_+, \quad (2)$$

where $\lambda, \mu_2 > 0$. Comparing with L_R loss, L_{RS} loss expects not only marginal discrimination between positive and negative triplets' scores, but also low scores for positive triplets.

Some other negative sampling losses for KG embedding models also try to make the discriminative marginal scores between the true and false triplets. HolE [48] suggests to use the margin-based ranking loss based on logistic function to discriminate the probabilities of true and false triplets. Trouillon and Nickel [54] present maximum likelihood of the logistic model which correspond to the softplus function as a differentiable alternative to margin-based ranking loss. ComplEx [31] defines a negative log-likelihood loss for modelling the discrimination. ProjE [30] also uses a negative sampling pointwise loss to rank the probability of true triples higher than the probability of false ones. RotatE [53] makes the true and false triplets as far away from a margin as possible. In addition to the negative sampling KG embedding methods, the neural network framework with cross-entropy loss and multi-label 1-k binary cross-entropy [49] [55], have been developed for KG embedding in recent years. In this paper, our work mainly focuses on improving the marginal ranking loss L_R and the combined loss L_{RS} for KG embedding.

3 The Proposed Methods

From the related works summarized in previous section, we find that most of the negative sampling KG embedding models are under the margin-based ranking loss L_R, which can not ensure low score for the golden triplet. L_R loss tries $f_r(h',t') - f_r(h,t) \geq \mu_1$, otherwise the parameters of the loss function will be updated. Although such loss can obtain a margin μ_1 between $f_r(h,t)$ and $f_r(h',t')$, but it may not ensure the score of correct triplet to be small enough to hold (h,r,t) [29]. Combined limit-based scoring loss L_{RS} framework [29] can improve the limit capacity for golden triplet greatly. However, margin-based ranking loss term $[\mu_1 + f_r(h,t) - f_r(h',t')]_+$ in L_{RS} is composed of $f_r(h,t)$ and $f_r(h',t')$, L_{RS} loss is still difficult to independently control the scores of negative triplets (h',r,t') without affecting that of positive ones. Moreover, both L_R and L_{RS} could not avoid the overlap of the positive and negative triplets' scores for complex relations completely.

In this section, we first present double limit scoring loss L_{SS} for optimizing KG embedding models, and give the metrics of our loss for optimization. Second upon L_{SS} loss framework, we present several KG embedding models, TransE-SS, TransH-SS, TransD-SS, ProjE-SS and ComplEx-SS. Third we introduce the optimization of our L_{SS} loss for the extended models, and at last the merits of our proposed loss are demonstrated by an intuitive numeric example.

3.1 Double Scoring Loss

A good loss framework for KG embedding tries to distinguish positive and negative triplet structures, meanwhile expects to learn the low score for correct triplet and the high score for incorrect triplet. In order to effectively distinguish positive and negative triplets through their scores, we propose the double scoring loss (denoted as L_{SS}), which consists of two scoring loss terms, and sets an upper bound for scoring positive triplets and a lower bound for scoring negative triplets separately. To this aim, we need to first define two limit-based scoring loss terms, $L_{S_{neg}}$ and $L_{S_{pos}}$.

$L_{S_{neg}}$ is the lower-limit scoring loss for negative triplets,

$$L_{S_{neg}} = \sum_{(h',r,t')\in\Delta'} [\mu_3 - f_r(h',t')]_+. \quad (3)$$

By the negative triplets (h',r,t'), $L_{S_{neg}}$ will be optimized to obtain higher scores than a given lower-limit μ_3, i.e. $f_r(h',t') \geq \mu_3$.

In contrast, we define the upper-limit scoring loss $L_{S_{pos}}$ for correct triplets,

$$L_{S_{pos}} = \sum_{(h,r,t)\in\Delta} [f_r(h,t) - \mu_2]_+, \quad (4)$$

$L_{S_{pos}}$ is also used in L_{RS} loss, and sets the upper bound μ_2 for the positive triplet's scores $f_r(h,t)$, i.e., $f_r(h',t') \leq \mu_2$.

Mathematically in all, the proposed double scoring loss is defined as:

$$L_{SS} = L_{S_{pos}} + \lambda L_{S_{neg}},$$

where $\lambda \geq 0$ is a parameter that balances the two limit-based scoring loss terms. A more detailed formula of the loss function is written as:

$$L_{SS} = \sum_{\substack{(h,r,t)\in\Delta \\ (h',r,t')\in\Delta'}} \{[f_r(h,t) - \mu_2]_+ + \lambda[\mu_3 - f_r(h',t')]_+\}. \tag{5}$$

Merits of Double Scoring Loss. Double scoring loss L_{SS} further improves combined limit-based scoring loss L_{RS} by replacing margin-based ranking loss term with the scoring loss term for negative triplet. The new loss dose not only have more flexible parameter tuning for negative triplets, but also restrict negative triplets without affecting positive ones directly. In details,

(1). Marginal scoring for positive and negative triplets, as an important goal of original margin-based ranking loss, can also be implemented by the double scoring loss. We usually set $\mu_3 > \mu_2$ for the double scoring loss aiming at making $f_r(h,t) \leq \mu_2$ and $f_r(h',t') \geq \mu_3$. It means that we expect the margin between $f_r(h,t)$ and $f_r(h',t')$ is at least $\mu_3 - \mu_2$.

(2). Ranking scores $f_r(h,t) \leq f_r(h',t')$, which is another aim of margin-based ranking loss, is also realized by the double scoring loss. If $f_r(h,t) \leq \mu_2$ and $f_r(h',t') \geq \mu_3$ by setting $\mu_3 > \mu_2$, obviously $f_r(h,t) < f_r(h',t')$.

(3). From the respect of avoiding overlaps between the score distributions of positive and negative, L_{SS} loss can ensure low score for positive triplets and high score for negative triplets by setting double scoring limit, i.e., small μ_2 and large μ_3.

(4). L_{SS} loss provides independent scoring limits for positive and negative triplets. Compared with L_R and L_{RS}, the mechanism makes the learning process of parameter more effective flexible, and the optimization for negative triplets more independent.

Comparing with the combined scoring loss L_{RS} [29], our loss further replaces the margin-based ranking loss item with a scoring loss for negative triplets, and have more direct scoring control for negative triplets.

3.2 Models

Upon the proposed double scoring loss, we extend TransE [18], TransH [19], TransD [20], ProjE [30] and ComplEx [31] separately to TransE-SS, TransH-SS, TransD-SS, ProjE-SS and ComplEx-SS. In this subsection, among the knowledge embedding models mentioned in related work, TransE and TransH have lower time complexities, and TransD, ProjE and ComplEx have much better predictive performances on KG completions. Our models, TransE-SS, TransH-SS and TransD-SS separately share the same scoring functions of TransE, TransH and TransD. ProjE-SS and ComplEx-SS extend the scoring by probability activation function based on the former models.

3.2.1 TransE-SS

Same to TransE, our TransE-SS regards each relation as a translation operation between the head embedding and tail embedding on the same vector space.

For a triplet (h,r,t), TransE-SS defines a scoring function $f_r(h,t)$ to measure its plausibility:

$$f_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|, \tag{6}$$

with restrictions $\|\mathbf{e}\|_2 = 1, \|\mathbf{r}\|_2 = 1$, where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in R^m$ are the embeddings of h,r,t respectively. This scoring function expects that the score is high for a positive, and low for a negative triplet.

Different from TransE that uses margin-based ranking loss L_R (Eq. (3)), TransE-SS uses the proposed double scoring loss L_{SS} (Eq. (5)) for optimization and training.

3.2.2 TransH-SS

TransE-SS is very efficient to 1-to-1 relation, but same to TransE, it still has the issues for N-to-1, 1-to-N and N-to-N relations. For example, by a 1-to-N relation, a head will only be translated to the same tail, that is, if r is a 1-to-N relation for $\{(h,r,t_i)\}_{i=1,2,\ldots,N}$, then $t_1 = t_2 = \ldots = t_N$, which does not handle the facts [19].

To address the issues of TransE-SS for complex relations, TransH-SS also uses the same scoring function as TransH [18], which models entities and relations as low-dimensional embedding vectors, and uses a hyperplane determined by the normal vector $\mathbf{w}_r \in R^m$ to translation operation.

For a triplet (h,r,t), TransH-SS first projects the head and tail entity embeddings $\mathbf{h}, \mathbf{t}$ into the hyperplane and gets the projections as

$$\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}^T\mathbf{h}\mathbf{w}$$
$$\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}^T\mathbf{t}\mathbf{w}.$$

Then TransH conducts the translation operation and defines the scoring function as

$$f_r(h,t) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|, \tag{7}$$

with restrictions $\|\mathbf{e}\|_2 \leq 1$, $|\mathbf{w}_r^T\mathbf{d}_r|/\|\mathbf{d}_r\|_2 \leq \epsilon$, $\|\mathbf{w}_r\|_2 = 1$.

Traditional TransH is based on the L_R loss framework, our TransH-SS optimizes the embeddings and model parameters by the new L_{SS} loss (Eq. (5)).

3.2.3 TransD-SS

TransD-SS defines the same scoring function as TransD [20], which takes the multiple types of entities and relations into account, and replaces transfer matrix in TransR by the product of two projection vectors of an entity-relation pair.

For a triplet (h,r,t), six vectors $\mathbf{h}, \mathbf{h}_p, \mathbf{r}, \mathbf{r}_p, \mathbf{t}, \mathbf{t}_p$ are used, where subscript p marks the projection vectors, $\mathbf{h}, \mathbf{h}_p, \mathbf{t}, \mathbf{t}_p \in R^m$ and $\mathbf{r}, \mathbf{r}_p \in R^n$. For each triplet (h,r,t), we set head and tail transfer matrices $\mathbf{M}_{rh}, \mathbf{M}_{rt} \in R^{m\times n}$ to project entities from entity space to relation spaces. They are defined as follows:

$$\mathbf{M}_{rh} = \mathbf{r}_p\mathbf{h}_p^T + \mathbf{I}^{n\times m}$$
$$\mathbf{M}_{rt} = \mathbf{r}_p\mathbf{t}_p^T + +\mathbf{I}^{n\times m}.$$

Then we define the scoring function as

$$f(h,r,t) = -log(\sigma(\mathbf{t}^T g(\mathbf{h} \oplus \mathbf{r}) + b_t))$$

where $\mathbf{h} \oplus \mathbf{r} = \mathbf{D}_h\mathbf{h} + \mathbf{D}_r\mathbf{r} + \mathbf{b}_c$ is the combination operator, and $\mathbf{D}_h, \mathbf{D}_r \in R^{n\times n}, \mathbf{b}_c \in R^n$.

$$f_r(h,t) = \|\mathbf{M}_{rh}\mathbf{h} + \mathbf{r} - \mathbf{M}_{rt}\mathbf{t}\| \tag{8}$$

with restriction $\|\mathbf{h}\|_2 \le 1, \|\mathbf{t}\|_2 \le 1,\ \|\mathbf{r}\|_2 \le 1,\ \|\mathbf{M}_{rh}\mathbf{h}\|_2 \le 1$, and $\|\mathbf{M}_{rt}\mathbf{t}\|_2 \le 1$.

TransD-SS can capture more power features than TransE-SS and TransH-SS. When the dimension $m = n$, and all projection vectors are set to zero, TransE-SS is equal to TransD-SS. Different from TransH-SS, TransD-SS considers the multiple types of entities and relations, which also determines the projection vectors simultaneously.

Similar to the former models, TransD-SS uses our double scoring loss L_{SS} to replace the L_R loss in TransD.

3.2.4 ProjE-SS

Different from the usual translation-based KG embedding model, HolE [48] and ProjE [30] take $P(Y_{hrt} = 1) = \sigma(\phi(r,h,t))$ as the probability scoring of (h,r,t). $Y_{hrt} \in \{-1,+1\}$ is the label for true or false triplet. ϕ is a measure function for an observed relation, and $\sigma(\cdot)$ is the activation function to further compute its probability to be correct.

In order to make the probabilities easier to be observed, $-log(\cdot)$ is suggested to be operated on the probabilities. For a triplet, the scoring function of ProjE-SS is defined as

$$f(h,r,t) = -log(\sigma(\mathbf{t}^T g(\mathbf{h} \oplus \mathbf{r}) + b_t)), \tag{9}$$

where $\mathbf{h} \oplus \mathbf{r} = \mathbf{D}_h\mathbf{h} + \mathbf{D}_r\mathbf{r} + \mathbf{b}_c$ is the combination operator, and $\mathbf{D}_h, \mathbf{D}_r \in R^{n\times n}, \mathbf{b}_c \in R^n$. The scoring function is a neural network operator, which takes two embedding vectors $(\mathbf{h},\mathbf{r})$ as input and creates a target vector through a combination operator, and then compute the probability scores of candidate embeddings for missing tail by a projection layer and activation function. The activation function can be logistic function or softmax function.

For the loss functions, HolE [48] suggests using the margin-based ranking L_R loss to rank the probability of true triples higher than the probability of false ones:

$$L = \sum_{(h,r,t)\in\Delta,(h',r,t')\in\Delta'} [\mu + \sigma(\phi(h',r,t')) - \sigma(\phi(h,r,t)]_+,$$

With the similar purpose, ProjE defines a pointwise loss [30]:

$$L = -\sum_t log(\sigma(\phi(h,r,t))) - \sum_{t'(t'\neq t)} log(1 - \sigma(\phi(h,r,t'))).$$

For ProjE-SS, we use the new L_{SS} loss framework to realize the optimization. Given an input (h,r), the loss function is defined upon all correct tails and m negative samples drawn from a negative candidate distribution $t' \sim E_{t'}$:

$$L_{SS} = \sum_t [f(h,r,t) - \mu_2]_+ + \lambda \sum_{t'(t'\neq t)} [\mu_3 - f(h,r,t')]_+.$$

3.2.5 ComplEx-SS

Same to ComplEx [31], ComplEx-SS considers low-rank matrix factorizations in complex space for scoring the triplets. For a relation, we can get a partially observed adjacency matrix $Y \in R^{n\times n}$, and each element $y_{h,t} \in \{-1,1\}$ denotes the true or false relation between entity pairs (h,t). If Y is low-sign-rank, then the sign-rank of it is the smallest rank of a real scoring matrix $X \in R^{n\times n}$ that has the same sign-pattern as Y [31]:

$$rank(Y) = min_{X\in R^{m\times n}}\{rank(Y)|sign(X) = Y\}.$$

By setting a low-rank $k \ll n$ on the scoring matrix X, it can be approximated by dot products of complex embeddings:

$$X = EW\overline{E}^T,$$

where $W \in C^{k\times k}$ is the diagonal matrix of eigenvalues, $E \in C^{n\times k}$ is the matrix of eigenvectors, and $\overline{E}$ is the complex conjugate of E. We also extract the real part of the decomposition as our scoring, as shown by [31].

$$X = Re(EW\overline{E}^T).$$

Individual relation scores X_{ht} between entities h and t can be predicted as $X = Re(e_h{}^T W \overline{e}_t)$ by the product of embeddings $e_h, e_t \in C^k$.

Such complex embedding model can effectively capture symmetric and antisymmetric relations, while the dot product is linearity in both space and time complexity. Factually in order to efficiently predict the scores X_r for the relations $r \in R$, our ComplEx-SS same to ComplEx [31] that introduces a relation complex vector $w_r \in C^k$ to define the measure function of (h,r,t):

$$\begin{aligned}\phi(h,r,t) =& Re(< w_r, e_h, \overline{e}_t >)\\ =& < Re(w_r), Re(e_h), Re(e_t) >\\ &+ < Re(w_r), Im(e_h), Im(e_t) >\\ &+ < Im(w_r), Re(e_h), Im(e_t) >\\ &- < Im(w_r), Im(e_h), Re(e_t) >,\end{aligned} \tag{10}$$

We further compute the probability of the true fact $r(h,t)$ is $P(Y_{rht} = 1) = \sigma(\phi(h,r,t))$, and then define the scoring function of ComplEx-SS as follows:

$$f_r(h,t) = -log(\sigma(\phi(h,r,t))). \tag{11}$$

Different to ComplEx that defines the loss function by minimize the negative log-likelihood of the model [31], our ComplEx-SS uses the double scoring loss for optimization:

$$L_{SS} = \sum_\Delta [f_r(h,t) - \mu_2]_+ + \lambda \sum_{\Delta'} [\mu_3 - f_r(h',t')]_+,$$

where Δ and Δ' denote the positive and negative triplets for a special relationship.

3.3 Optimization and Training

The optimization for minimizing the double limit scoring loss with the constraints mentioned above is carried out gradient descent [56] over the possible entities, translation vectors and other parameters. When a golden triplet is visited, a negative triplet is randomly constructed according to the reference [19]. Given a mini-batch of training triplets $\{(h_i,r_i,t_i)\}_{i=1\sim N_B}$, the double limit scoring loss of a mini-batch is

$$L_{SS} = \sum_{i=1}^{N_B} L_{SS}(i), \tag{12}$$

where $L_{SS}(i) = L_{S_{pos}}(i) + \lambda L_{S_{neg}}(i) = [f_{r_i}(h_i, t_i) - \mu_2]_+ + \lambda[\mu_3 - f_{r_i}(h_i', t_i')]_+$. The gradient L_{SS} of a mini-batch is generated from the stochastic gradient of each pair of positive and negative triplets $\{(h_i, r_i, t_i), (h_i', r_i, t_i')\}$ as follows:

$$\nabla L_{SS} = \sum_{i=1}^{N_B} \nabla L_{SS}(i) \tag{13}$$

where $\nabla L_{SS}(i) = \nabla L_{S_{pos}}(i) + \lambda \nabla L_{S_{neg}}(i) = \nabla[f_{r_i}(h_i, t_i) - \mu_2]_+ + \lambda \nabla[\mu_3 - f_{r_i}(h_i', t_i')]_+$. Specifically, if $f_{r_i}(h_i, t_i) - \mu_2 > 0$, $\nabla L_{S_{pos}} = \nabla f_{r_i}(h_i, t_i)$, else $\nabla L_{S_{pos}} = 0$. If $\mu_3 - f_{r_i}(h_i', t_i') > 0$, $\nabla L_{S_{neg}} = -\nabla f_{r_i}(h_i', t_i')$, else $\nabla L_{S_{neg}} = 0$.

Comparing with L_{RS}, L_{SS} can optimize negative triplets more independently. The major differences between L_{SS} and L_{RS} are that $[\mu_1 + f_r(h,t) - f_r(h',t')]_+$ in L_{RS} and $[\mu_3 - f_r(h',t')]_+$ in L_{SS}. From the view of computation, $\nabla[\mu_3 - f_r(h',t')]_+$ can optimize negative triplets more independently than $\nabla[\mu_1 + f_r(h,t) - f_r(h',t')]_+$. Because that for $[\mu_1 + f_r(h,t) - f_r(h',t')]_+$, once $\mu_1 + f_r(h,t) - f_r(h',t') > 0$, the negative score $f_r(h',t')$ is needed to be updated by $\nabla f_r(h',t')$ from $\nabla[\mu_1 + f_r(h,t) - f_r(h',t')]$, the corresponding positive score $f_r(h,t)$ is also updated by $\nabla f_r(h,t)$ from that. But for $[\mu_3 - f_r(h',t')]_+$, the update of negative score $f_r(h',t')$ by $\nabla[\mu_3 - f_r(h',t')]$ will not affect positive score $f_r(h,t)$. Such independent parameter control mechanism for positive and negative scores by L_{SS} also makes L_{SS} converge faster than L_{RS} (see Fig. 6 in Section 4.3).

Algorithm 1. KG embedding models with L_{SS}

Input: Positive training triplets $\Delta = \{(h,r,t) | h,t \in E, r \in R\}$, E and R are respectively the set of entities and relations. Negative training triplets $\Delta' = \emptyset$.
Output: Entity and relation embedding $\mathbf{E}$ and $\mathbf{R}$
Stage 1: Initialization of Knowledge Graphs.
Entity embedding $\mathbf{E} \longleftarrow \text{initialization}(N_e, m)$
Relation embedding $\mathbf{R} \longleftarrow \text{initialization}(N_r, n)$
// initialization(a, b) produces a matrix with size [a, b] by initialized randomly or the results of TransE [18]
Stage 2: Construct Negative Triplets.
for each (h,r,t) in positive sample set Δ **do**
  $(h',r,t') = \text{generate\_negative}((h,r,t))$
  // using *unif/bern* strategy in [19] or *candidate distribution* strategy in [30] for generating negative samples.
  $\Delta' = \Delta' \cup \{(h',r,t')\}$
end for
Stage 3: Learning Embeddings of Entities and Relations.
Loop:
P = sample_batch(Δ, Δ', B) // sample a mini-batch of size B at random from positive and negative training samples.
for each $\{(h,r,t),(h',r,t')\}$ in P **do**
  calculate $\nabla[f_r(h,t) - \mu_2]_+ + \lambda\nabla[\mu_3 - f_r(h',t')]_+$
  update embeddings $\mathbf{E}$ and $\mathbf{R}$
end for
return $\mathbf{E}, \mathbf{R}$

In training stage, for TransE-SS, TransH-SS, TransD-SS and ComplEx-SS, we use the same negative sampling method as the former models [19], [20], [31]. For ProjE-SS, we draw m negative samples from a negative candidate distribution $t' \sim E_{t'}$. The embeddings of TransE-SS, TransH-SS, ComplEx-SS and ProjE-SS are initialized randomly. TransD-SS is same to TransD [20] that uses the results of TransE as initialized embeddings. The detailed training process of our proposed knowledge graph embedding models with double scoring loss L_{SS} is illustrated in Algorithm 1.

We also give the parameter and operation complexities of our presented models and related methods in Table 1, where m and n are the dimension of the entity and relation embedding spaces, N_e, N_r and N_t are the number of entities, relations and triplets in knowledge graphs, θ is the sparse degree of matrix, c is the number of filter parameters in convolutional operation and l is the length of paths in knowledge graphs. Compared with former models, our models do not increase complexities of the parameter and operation. TransE-SS, TransH-SS, TransD-SS, ProjE-SS and ComplEx-SS separately have the same parameters and complexities as the former models.

3.4 Examples Analysis

In order to analyze the effectiveness of our double loss L_{SS} for KG embedding models intuitively, we compare TransE-SS with TransE (L_R loss) and TransE-RS (L_{RS} loss) by an intuitive example of optimizing positive and negative triplet's scores.

Given some pairs of correct and incorrect triplets with the initialized scores $[f_r(h,t), f_r(h',t')]$ as $[0.5, 1.5]$, $[0.5, 2.5]$, $[0.5, 3.5]$, $[3.5, 4.5]$, $[3.5, 5.5]$, $[3.5, 6.5]$, $[4.5, 5.5]$, $[4.5, 6.5]$, $[4.5, 7.5]$, $[6.5, 7.5]$, $[6.5, 8.5]$, $[6.5, 9.5]$, shown in Fig. 2a, where a line connects a pair of scores $[f_r(h,t), f_r(h',t')]$. We assume that all the pairs of scores are from a same 1-to-n relation category, that is, the task is to predict the missing tails with the fixing head and relation $(h, r, ?)$. We can see a large overlapping between the positive and negative triplets' scores, and there are three kinds of scores for the pairs: the first five pairs are the $\{low f_r(h,t), low f_r(h',t')\}$ case, the sixth pair is the $\{low f_r(h,t), high f_r(h',t')\}$ case, and the last six belong to the $\{high f_r(h,t), high f_r(h',t')\}$ case. We will test the three losses L_R, L_{RS} and L_{SS} on these pairs of scores separately, and rank all the candidate tails t^* (including correct tails t and incorrect tails t') by their scorings $f(h,r,t^*)$ in ascending order. The ideal result should be that positive tails are in front of negative tails: $t, t, \ldots, t, t', t', \ldots, t'$.

The parameters used are: $\mu_1 = 2$ for loss L_R, $\{\mu_1 = 2, \mu_2 = 4, \lambda = 1\}$ for loss L_{RS}, $\{\mu_2 = 4, \mu_3 = 6, \lambda = 1\}$ for loss L_{SS}, the learning rate $\alpha = 0.1$ for all the losses. Figs. 2b, 2c and 2d shows the results of scores after gradient L_R, L_{RS} and L_{SS} respectively.

Margin-based Ranking Loss. For the results of margin-based ranking loss $max(0, \mu + f_r(h,t) - f_r(h',t'))$, as illustrated in Fig. 2b, although with the margin parameter $\mu_1 = 2$, all the pairs satisfy that $f_r(h',t') - f_r(h,t) > \mu_1$, there are still the following issues. (1) Margin-based ranking loss cannot ensure that the positive triplet's score $f_r(h,t)$ is within a low value domain. For example, the positive triplets' scores in last five pairs are all larger than 4, which is not good for entity prediction according to the scoring function. (2) There still three kinds of score cases $\{low f_r(h,t), low f_r(h',t')\}$, $\{low f_r(h,t), high f_r(h',t')\}$ and $\{high f_r(h,t), high f_r(h',t')\}$ shown as Fig. 2b, although the seventh pair are optimized into normal

TABLE 1
Summary of Related Models

Model	#Parameter	#Time Complexity
LFM	$O(N_em+N_rn^2)(m=n)$	$O((m^2+m)N_t)$
SE	$O(N_em+2N_rn^2)(m=n)$	$O(2m^2N_t)$
SME(LIN)	$O(N_em+N_rn+4mk+4k)(m=n)$	$O(4mkN_t)$
SME(BILIN)	$O(N_em+N_rn+4mks+4k)(m=n)$	$O(4mksN_t)$
UM	$O(N_em)$	$O(N_t)$
TransR	$O(N_em+N_r(m+1)n)$	$O(2mnN_t)$
TranSparse	$O(N_em+N_r(1-\theta))mn$	$O(mnN_t)$
STransE	$O(N_em+2N_r(n^2+n))$	$O(2m^2N_t)$
TransG	$O(N_em+N_rnc)$	$O(mcN_t)$
KG2E	$O(N_em+N_rn)$	$O(mN_t)$
SLM	$O(N_em+N_r(2k+2nk))$	$O((2mk+k)N_t)$
NTN	$O(N_em+N_r(n^2s+2ns+2s))$	$O(((m^2+m)s+2mk+k)N_t)$
RESCAL	$O(N_em+N_rn^2)$	$O(m^2N_t)$
DistMult	$O(N_em+N_rn)$	$O(mN_t)$
HolE	$O(N_em+N_rn)$	$O(mlog(m)N_t)$
ConvE	$O(N_em+N_rn)$	$O(N_tc)$
PTransE	$O(N_em+N_rn)$	$O(lN_t)$
RTransE	$O(N_em+N_rn$	$O(lN_t)$
TransE	$O(N_em+N_rn)$	$O(N_t)$
TransE-RS	$O(N_em+N_rn)$	$O(N_t)$
TransE-SS	$O(N_em+N_rn)$	$O(N_t)$
TransH	$O(N_em+2N_rn)$	$O(2mN_t)$
TransH-RS	$O(N_em+2N_rn)$	$O(2mN_t)$
TransH-SS	$O(N_em+2N_rn)$	$O(2mN_t)$
TransD	$O(2N_em+2N_rn)$	$O(2nN_t)$
TransD-SS	$O(2N_em+2N_rn)$	$O(2nN_t)$
ProjE	$O(N_em+N_rn+5m)$	$O(2(m^2+m)N_t)$
ProjE-SS	$O(N_em+N_rn+5m)$	$O(2(m^2+m)N_t)$
ComplEx	$O(N_em+N_rn)$	$O(mN_t)$
ComplEx-SS	$O(N_em+N_rn)$	$O(mN_t)$

$\{lowf_r\ (h,t), highf_r(h',t')\}$ case. The scores of positive and negative triplets have a great overlapping, which lead to the ranking of the correct and incorrect are confused.

Combined Limit-Based Scoring Loss. The combined limit-based scoring loss L_{RS} enforces an upper-limit score $\mu_2 = 4$ for all positive triplets, which means that $f_r(h,t) < \mu_2$ is also expected. As Fig. 2c shown, L_{RS} dose not only guarantee the discrimination between scores of positive and negative triplets at least $\mu_1 = 2$, but also ensures that score of positive triplet is within an expected low value domain $\mu_2 = 4$. Thus, L_{RS} can effectively avoid $\{highf_r(h,t), highf_r(h',t')\}$ case, and the last five $\{highf_r(h,t), highf_r(h',t')\}$ pairs can be optimized into the $\{lowf_r(h,t), highf_r(h',t')\}$ case. However L_{RS} still can not solve $\{lowf_r(h,t), lowf_r(h',t')\}$ case, and there maybe partial overlapping when the initial scores of positive and negative triplets are both lower, e.g., the first five pairs.

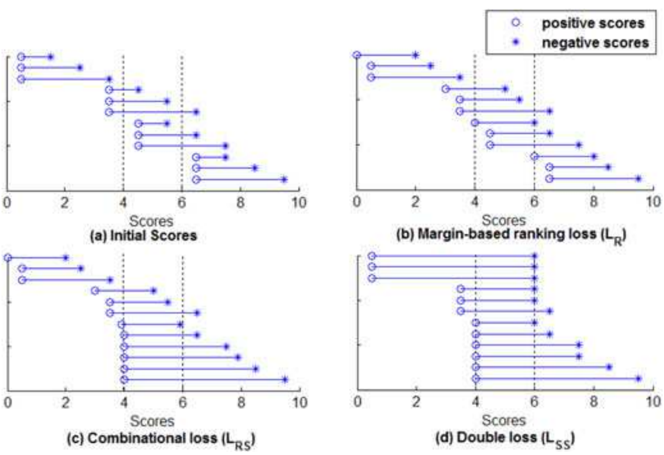


Fig. 2. Instances on scores of triplets.

Double Limit Scoring Loss. Comparing with L_R and L_{RS}, our double scoring loss can completely avoid overlapping between scores of positive and negative triplets, and also can provide two separated optimization processes for positive and negative triplets. L_{SS} simultaneously sets an upper-limit score μ_2 for positive triplets and a lower-limit μ_3 for negative triplets, where $\mu_3 - \mu_2$ imply the margin between the pairs of scores. As illustrated in Fig. 2d, all positive triplets have scores lower than $\mu_2 = 4$ and negative triplets have scores larger than $\mu_3 = 6$, and there is no overlap between scores of positive and negative triplets.

4 Experiments

We compare our proposed L_{SS} based models with other KG embedding models for link prediction [18] and triplet classification [17] tasks on two popular KGs, FreeBase [33] and WordNet [32]. Freebase contains a large number of world facts, and WordNet is a large lexical knowledge graph. In our experiments, we use some subsets of the two KGs, and the statistics of the subsets are given in Table 2. WN18, WN18RR and WN11 are from WordNet, and FB15k, FB15K-237 and FB13 are from Freebase. Earlier compared WN18 and FB15K [18] both include a large number of inverse

TABLE 2
Experimental Datasets

Dataset	#Rel	#Ent	#Train	#Valid	#Test
WN18	18	40,943	141,442	5,000	5,000
FB15K	1,345	14,951	483,142	50,000	59,071
WN18RR	11	40,943	86,835	3034	3134
FB15K-237	237	14,541	272,115	17,535	20,466
WN11	11	38,696	112,581	2,609	10,544
FB13	13	75,043	316,232	5908	23,733

triplets, that is, the test set frequently contains triples such as (s, r_1, o) while the training set contains its inverse (o, r_2, s). So FB15K-237 [57] and WN18RR [49] were introduced to remove the inverse relations and reclaimed as new datasets for current KG embedding evaluations.

4.1 Link Prediction

Link predictions [27], [28] aim to predict the missing head $(?, r, t)$ or tail $(h, r, ?)$ in the relation facts. For a testing triplet (h, r, t), every entity in the KG will replace the missing entity to construct the predicted triplets, and then such triplets are ranked in descending order according to the scores by scoring function. Based on the score rank, several metrics are usually reported: mean rank (MR) of correct entities, mean reciprocal rank (MRR) of correct entities, and the proportion of top-k rank (Hits@k) for correct entities. A good model should have low "MR", high "MRR" and high "Hits@10". For constructing the corrupted triples, "unif" denotes the traditional way of replacing head or tail with equal probability, and "bern" denotes reducing false negative labels by replacing head or tail with different probabilities following [19]. The settings "raw" and "filt" for the metrics distinguish whether or not to consider the impact of a corrupted triplet existing in the correct KG.

4.1.1 Results on WN18 and FB15K

First, we follow the experimental procedures and metrics of the most negative sampling KG embedding models (such as translation models [18], [19] etc.) for our evaluations on WN18 and FB15K. In this experiment, two metrics (MR and Hits@10) are reported.

Parameters Settings. We comparing the series of TransE, TransH, TransD, ProjE and ComplEx with different losses, and run them by our code framework.[1] We search the best settings of these models from the following parameters: learning rate α from {0.0005, 0.001, 0.005, 0.01}, the embedding dimension m from {50, 80, 100, 150, 200}, the batch size B from {50, 75, 100, 120, 200, 480, 500, 960, 1000, 1200, 4800}, margin size μ_1 from {0.1, 0.25, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}. $\{L_1, L_2\}$ distances for loss functions, and weight parameter λ from {0.05, 0.1, 0.5, 1, 2, 3, 10, 25} for L_{RS} and L_{SS}. For TransE-SS, TransH-SS and TransD-SS, upper limit μ_2 score for positive triplets from {0.25, 1, 2, 3, 4, 5, 6, 7, 8, 10, 15}, and lower limit μ_3 score for negative triplet from {μ_2 + {0.1, 0.25, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}}. Parameter C for TransH series from {0.0005, 0.0625, 0.25, 1.0}. For ComplEx-SS, upper limit μ_2 score for positive triplets is $-log(p_+)$, p_+ from $\{0.3, 0.4, 0.6, 0.7, 0.8, 0.85, 0.9, 0.95, 0.99\}$, and lower limit score μ_3 for negative triplets is $-log(p_-)$, p_- from $\{0.7, 0.6, 0.4, 0.3, 0.2, 0.15, 0.05, 0.001\}$. For ProjE, upper limit μ_2 is $-log(p_+/n_+)$, and lower limit score μ_3 is $-log(p_-/n_-)$, n_+ and n_- are respectively the number of correct and incorrect candidates for a given inputs. The dropout rate is 0.2 on WN18 and FB15K, and negative sampling rate p_d is from $\{0.1, 0.15, 0.20, 0.25, 0.5\}$. The optimal parameters are determined by the validation set. We traverse all the training triplets for 1000 rounds. The optimal configurations of the models on link prediction is illustrated in Table 4.

Table 3 shows the evaluation results on two datasets WN18 and FB15K, and the top-3 results in each column with bold marker are given. The original results of the series of TransE, TransH, TransD, ComplEx and ProjE from the references [18], [19], [20], [31], [29] are given in the round brackets. For the other compared models, we report the original results from [34], [21], [49], [48], [47], [35].

From Table 3, we can see that: (1) All the proposal models with L_{SS} loss outperform the corresponding former models with L_R and L_{RS} on all the metrics, which indicates that L_{SS} loss can boost the capability of finding missing facts and completing KGs. Detailed improved results for Hit@10 (bern, filt) metric are as follows. On WN18, the results are about increased by TransE-SS 0.5%, TransH-SS 0.6%, TransD-SS 0.3%, ComplEx-SS 0.5% and ProjE-SS 0.4% than corresponding L_{RS} loss models. On FB15K, the results are increased by TransE-SS 1.8%, TransH-SS 4.8%, TransD-SS 5.5%, ComplEx-SS 1.1% and ProjE-SS 0.1%. (2) Among all the compared methods, TransE-SS, TransH-SS and TransD-SS have remarkable improvements (especially on FB15K), and meanwhile have lower computation complexities. (3) From the metric of Hit@10, ComplEx-SS and ProjE-SS achieve more better performances on the link prediction tasks.

4.1.2 Results on WN18RR and FB15K-237

We further evaluate our proposed models for link predictions on WN18RR and FB15K-237. As WN18 and FB15K both include a large number of inverse relations like $\{(s, r_1, o), (o, r_2, s)\}$ [57], [49], many test triplets (s, r, o) can be inferred via a directly linked triplet (s, r_1, o) or (o, r_2, s).

FB15K-237 [57] and WN18RR [49] are two more challenging datasets for KG completions, where the inverse relations are deleted and the main relation patterns are symmetry/antisymmetry and composition patterns. In recent years, many non-translation models like DisMult [47], ComplEx [31] and ConvE [49] etc. are tested on FB15K-237 and WN18RR by five metrics, MR, MRR, Hits@1, Hits@3 and Hits@10. In this experiment, by the five metrics, we compare our TransE-SS, TransH-SS, TransD-SS, ProjE-SS and ComplEx-SS with their former loss models [18], [19], [20], [30], [31] and some baseline models Rescal [37], Discult [47] and ConvE [49]. We report the results of DisMult and ConvE from the references [47], [49], and rerun the other compared models by the released codes from [18], [19], [20], [31], [30]. We evaluate the models in the "bern" and "filt" setting.

Parameters Settings. We search the best settings from the same parameter fields as the former experiments on WN18

1. https://github.com/IIE-UCAS/Knowledge-Embedding-with-Double-Loss

TABLE 3
Evaluation Results on Link Prediction

Models	WN18				FB15k			
	Mean		Hits@10 (%)		Mean		Hits@10(%)	
	raw	filt	raw	filt	raw	filt	raw	filt
Unstructured [27]	315	304	35.5	38.2	1074	979	4.5	6.3
SE [24]	1,011	985	68.5	80.5	273	162	28.8	39.8
LMF [26]	469	456	71.4	81.6	283	164	26.0	33.1
RESCAL [37]	1,180	1,163	37.2	52.8	828	683	28.4	44.1
SME(linear) [27]	545	533	65.1	74.1	274	154	30.7	40.8
SME(bilinear) [27]	526	509	54.7	61.3	284	158	31.3	41.3
TransR(unif) [34]	232	219	78.3	91.7	226	78	43.8	65.5
TransR(bern) [34]	238	225	79.8	92.0	198	77	48.2	68.7
TransSparse(unif) [21]	233	221	79.6	93.4	216	66	50.3	78.4
TransSparse(bern) [21]	223	211	80.1	93.2	190	82	53.7	79.9
STransE(unif) [35]	224	211	80.8	93.2	220	69	51.5	78.4
STransE(bern) [35]	219	206	80.9	93.4	219	68	51.6	79.7
DistMult [47]	987	902	79.2	93.6	224	97	51.8	82.4
HolE [48]	387	361	80.4	94.9	209	75	54.9	73.9
TransE(unif) [18]	288 (263)	265 (251)	80.0 (75.4)	94.1 (89.2)	177 (243)	63 (125)	46.4 (34.9)	63.8 (47.1)
TransE(bern) [18]	291	282	81.4	94.6	198	103	49.8	65.8
TransE-RS(unif) [29]	289 (362)	281 (348)	81.0 (80.3)	94.1 (93.7)	176 (161)	**41** (62)	52.6 (53.1)	74.8 (72.3)
TransE-RS(bern) [29]	278 (385)	265 (371)	82.0 (80.4)	94.5 (93.7)	159 (161)	57 (63)	53.8 (53.2)	74.7 (72.1)
TransE-SS(unif)	285	279	83.1	94.4	170	**39**	54.3	78.7
TransE-SS(bern)	276	263	83.6	95.0	**155**	54	**55.8**	76.5
TransH(unif) [19]	347 (318)	331 (303)	80.3 (75.4)	94.5 (86.7)	212 (211)	84 (84)	44.1 (42.5)	62.1 (58.5)
TransH(bern) [19]	301 (401)	286 (388)	80.6 (73.0)	94.7 (82.3)	306 (212)	211(87)	44.8 (45.7)	58.8 (64.4)
TransH-RS (unif) [29]	**201** (401)	**188** (389)	80.2 (81.2)	94.7 (94.7)	210 (163)	55 (64)	51.9 (53.4)	81.3 (72.6)
TransH-RS(bern) [29]	214 (371)	202 (357)	81.0 (80.3)	94.5 (94.5)	179 (178)	62 (77)	53.4 (53.6)	78.7 (75.0)
TransH-SS(unif)	**182**	**170**	81.8	95.1	**166**	54	**55.3**	82.5
TransH-SS(bern)	**184**	**173**	82.1	95.1	177	61	54.6	83.5
TransD(unif) [20]	259 (242)	246 (229)	81.5 (79.2)	94.8 (92.5)	219 (211)	70 (67)	50.1 (49.4)	77.1 (74.2)
TransD(bern) [20]	283 (224)	270 (212)	82.0 (79.6)	95.0 (92.2)	194(194)	89 (91)	54.0 (53.4)	78.4 (77.3)
TransD-SS(unif)	267	250	**83.0**	95.0	201	70	53.8	82.0
TransD-SS (bern)	248	237	**83.1**	**95.3**	176	69	**55.3**	83.9
ComplEx [31]	466	451	81.5	**95.4** (94.7)	215	92	54.4	84.8 (84.0)
ComplEx-SS	431	418	**84.0**	**95.9**	179	53	53.8	**85.9**
ProjE [30]	262	243	81.7	95.0	148 (124)	45 (34)	55.0 (54.7)	**87.8** (88.4)
ProjE-SS	282	265	82.6	**95.4**	**158**	**37**	**55.3**	**87.9**

and FB15K. L_1 distance for the series of TransE and TransH, and L_2 distance for the series of TransD. For ProjE and ProjE-SS, dropout rate is 0.2 on WN18RR and 0.5 on FB15K-237. We traverse all the training triplets for 3000 rounds. The optimal configurations of our presented models on link prediction is illustrated in Table 6.

The experimental results on FB15K-237 and WN18RR are given in Table 5, where the top-3 results in each column with bold marker are given.

From Table 5, we can see that: (1) Our presented models with L_{SS} loss outperform the corresponding former models with L_R and L_{RS} on all the metrics. The results also prove the effectiveness of our L_{SS} loss. (2) Among all the compared methods, TransD-SS performs best among the compared models on FB15K-237, and achieves highest Hits@10 51.0%, Hits@3 35.6%, and MRR 32.7%. ComplEx-SS performs well on WN18RR, and obtains the highest Hits@10 50.6% and Hits@3 44.5%. ConvE shows the best prediction capability on Hits@1 metric on WN18RR and FB15K-237. (3) We can see that the translation models with L_{RS} and L_{SS} generally have remarkable improvements on all the metrics. However, similar to the L_R loss, they still does not perform well on Hits@1 metric of WN18RR. The reason maybe that the main relation patterns is the symmetry pattern in WN18RR, the scoring functions of the translation models do not well on it. Comparing to translation models, the score functions of ComplEx can better model the the symmetry pattern, thus our extended ComplEx-SS also can achieve better prediction on this metric. (4) Comparing the results of Tables 3 and 4, we can find that, the improvements of the translation models with L_{SS} on new datasets (WN18RR and FB15K-237) are not more significants than that on the old datasets (WN18 and FB15K). By contrast, ComplEx with L_{SS} can obtain remarkable improvements on both old and new datasets. Therefore, the ability of a model to process a certain pattern (such as symmetry/antisymmery) is also related to the scoring function. So we can not directly conclude that L_{SS} is more prone to symmetry/antisymmery or inverse relationship.

TABLE 4
Parameter Configurations for WN18 and FB15K

WN18	α	μ_1	μ_2	μ_3	m	B	λ	C/p_d
TransE	0.01	4	-	-	50	75	-	-
TransE-RS	0.01	4	4	-	50	75	1	-
TransE-SS	0.01	-	4	8	50	75	1	-
TransH	0.01	4	-	-	100	120	-	0.0005
TransH-RS	0.01	4	3	-	100	120	1	0.0005
TransH-SS	0.01	-	3	7	100	120	1	0.0005
TransD	0.01	4	-	-	100	1200	-	-
TransD-SS	0.01	-	4	8	100	1200	3	-
ComplEx	0.01	-	-	-	200	100	-	-
ComplEx-SS	0.01	-	0.8	0.4	200	100	1	-
ProjE	0.01	-	-	-	200	100	-	$p_d = 0.1$
ProjE-SS	0.01	-	0.99	0.01	200	100	1	$p_d = 0.1$
FB15K	α	μ_1	μ_2	μ_3	m	B	λ	C/p_d
TransE	0.001	4	-	-	100	120	1	-
TransE-RS	0.001	2	6	-	100	960	1	-
TransE-SS	0.001	-	6	8	100	960	1	-
TransH	0.001	2	-	-	100	960	-	0.0625
TransH-RS	0.001	1	7	-	100	960	1	0.0625
TransH-SS	0.001	-	7	8	100	960	3	0.0625
TransD	0.001	1	-	-	100	960	-	-
TransD-SS	0.001	-	8	9	100	960	2	-
ComplEx	0.01	-	-	-	200	100	-	-
ComplEx-SS	0.01	-	0.9	0.7	200	100	1	-
ProjE	0.01	-	-	-	200	100	-	$p_d = 0.25$
ProjE-SS	0.01	-	0.99	0.01	200	100	-	$p_d = 0.25$

4.2 Triplet Classification

Triplet classification is a binary classification problem to decide whether a given triplet (h, r, t) is correct or not. This task is usually tested by translation models and semantic matching models with margin-based ranking loss, but it is rarely validated by nonlinear models like ConvE and ProjE. So in this experiment we only test the series of the compared translation models. We use three datasets, WN11, FB13 and FB15K (see Table 2) for the experiment. The training procedures are the same as the experiments of link predictions. For a testing triplet (h, r, t), it will be predicted positive if the score $f_r(h, t)$ is below a relation-specific threshold, otherwise negative. The relation-specific threshold is optimized by maximizing classification accuracies on the validation set.

We compare our models TransE-SS, TransH-SS and TransD-SS with baseline methods reported in [19], [20], [29], [34] who used the same data sets. In this experiment, the procedure of searching parameters and training procedures are same as former link predictions. We traverse all the training triplets for 1000 rounds. In the test phase, we need negative triples for the binary classification evaluation. The data sets WN11 and FB13 released by NTN [17] already have negative triples. For FB15k, we construct the negative triplets following [17]. For a testing triplet (h, r, t), it will be predicted positive if the score $f_r(h, t)$ is below a relation-specific threshold, otherwise negative. The relation-specific threshold is optimized by maximizing classification accuracies on the validation set. The optimal parameter configurations of our extended models on triplet classification is given in Table 7.

The experimental results on triplet classification are shown in Table 8. On WN11 our three models all can reach accuracy of more than 86%, and TransD-SS can achieve the highest accuracy 86.8%. On FB13, our three models is comparable to former loss models. On FB15K, our models have significant improvement compared to former models, and TransD-SS still performs best resulting 92.3% accuracy among the compared models.

4.3 Discussion

Our proposed double scoring loss framework enforces an upper-limit for positive triplets and a lower-limit for

TABLE 5
The results of entity prediction on WN18RR and FB15K-237

Models	WN18RR					FB15K-237				
			Hits(%)					Hits(%)		
	MR	MRR(%)	@1	@3	@10	MR	MRR(%)	@1	@3	@10
RESCAL [47]	10077	24.7	19.9	27.7	35.2	508	22.1	13.9	24.3	39.2
DisMult [47]	5110	**43**	**39**	**44**	**49**	254	24.1	15.5	26.3	41.9
ConvE [49]	5277	**46**	**39**	43	48	246	**31.6**	**23.9**	**35.0**	**49.1**
TransE [18]	3530	20.7	2.2	36.1	47.8	189	27.9	19.3	30.5	44.9
TransE-RS [29]	3415	20.8	2.3	36.3	47.8	177	28.2	19.4	31.2	46.1
TransE-SS (this paper)	3199	20.9	2.5	37.1	47.9	172	28.4	19.6	31.7	47.0
TransH [19]	3972	19.8	0.7	36.3	46.3	218	26.7	17.7	29.9	44.5
TransH-RS [29]	3421	18.1	0.9	36.9	47.6	207	27.3	17.6	30.6	46.4
TransH-SS (this paper)	3242	20.1	1.0	37.3	47.8	200	28.5	17.8	31.2	46.7
TransD	**2562**	20.3	4.2	35.5	46.0	**134**	30.5	20.5	31.7	47.7
TransD-RS	**2403**	21.0	3.7	35.6	47.1	**127**	**31.8**	**23.1**	**35.5**	**50.3**
TransD-SS (this paper)	**2392**	23.6	4.3	35.8	**49.6**	**114**	**32.7**	**23.2**	**35.6**	**51.0**
ComplEx [31]	5246	40.1	36.2	**42.5**	47.1	305	24	15.2	26.4	42.3
ComplEx-SS (this paper)	5152	**41.3**	**37.8**	**44.5**	**50.6**	301	24.7	15.7	27.3	43.4
ProjE [30]	3639	40.2	33.7	41.4	48.6	179	30.7	21.5	33.7	**49.1**
ProjE-SS (this paper)	3599	41.0	33.8	41.9	48.8	178	30.9	21.8	34.1	**49.1**

TABLE 6
Parameter Configurations for WN18RR and FB15K-237

WN18RR	α	μ_1	μ_2	μ_3	m	B	λ	C/p_d
TransE	0.0005	5	-	-	50	100	-	-
TransE-RS	0.0005	5	15	-	50	100	1	-
TransE-SS	0.0005	-	3	8	50	50	1	-
TransH	0.0005	5	-	-	100	480	-	0.0625
TransH-RS	0.0005	5	4	-	100	480	1	0.0625
TransH-SS	0.0005	-	4	9	100	480	1	0.0625
TransD	0.0005	1	-	-	100	1000	-	-
TransD-RS	0.0005	1	1	-	100	1000	3	-
TransD-SS	0.0005	-	1	2	100	1000	25	-
ComplEx	0.005	-	-	-	200	500	0.05	-
ComplEx-SS	0.005	-	0.95	0.6	200	500	0.05	-
ProjE	0.01	-	-	-	200	200	-	p_d = 0.15
ProjE-SS	0.01	-	0.95	0.05	200	200	1	p_d = 0.15
FB15K-237	α	μ_1	μ_2	μ_3	m	B	λ	C/p_d
TransE	0.0005	5	-	-	150	500	-	-
TransE-RS	0.0005	5	8	-	150	500	0.1	-
TransE-SS	0.0005	-	7	10	150	500	1	-
TransH	0.001	5	-	-	100	480	-	0.0625
TransH-RS	0.0005	2	5	-	100	480	1	0.0625
TransH-SS	0.0005	-	5	7	100	480	1	0.0625
TransD	0.0005	1	-	-	100	1000	-	-
TransD-RS	0.005	1	2	-	100	1000	1	-
TransD-SS	0.005	-	1	1.5	100	1000	10	-
ComplEx	0.005	-	-	-	80	1200	-	-
ComplEx-SS	0.005	-	0.95	0.6	80	1200	0.05	-
ProjE	0.01	-	-	-	200	200	-	p_d = 0.25
ProjE-SS	0.01	-	0.95	0.05	200	500	1	p_d = 0.25

negative triplets, which can avoid overlaps between score distributions of positive and negative triplets. Also, there are several parameters including upper-limit μ_2 for scoring positive triplets, lower-limit μ_3 for scoring negative triplets, combinational weight λ for a pair of scores, in our models. In the following, we will test the score distributions of positive and negative triplets, and explore how different parameters affect the performance of our models.

4.3.1 Score Distributions

We test the positive and negative triplets' score distributions by different models, Trans(E,H), Trans(E,H)-RS, and Trans(E,H)-SS.

TABLE 7
Parameter Configurations for Triplet Classification

Models	Datasets	α	μ_2	μ_3	m/n	B	λ	C/p_d
TransE-SS	WN11	0.01	2	13	100	120	1	-
	FB13	0.001	5	6	100	480	1	-
	FB15k	0.001	4	5	100	960	1	-
TransH-SS	WN11	0.01	2	12	100	120	1	0.0005
	FB13	0.01	5	8	100	1200	1	0.0625
	FB15k	0.001	7	8	100	960	3	-
TransD-SS	WN11	0.1	2	10	100	1200	1	-
	FB13	0.01	6	8	100	1200	1	-
	FB15k	0.01	8	9	100	960	2	-

TABLE 8
Accuracies(%) on Triplets Classification

Dataset	WN11	FB13	FB15K
RESCAL [37]	50.2	61.5	51.0
SE [24]	53.0	75.2	65.4
LMF [26]	73.8	84.3	68.3
SME(linear) [27]	68.4	62.8	69.7
SME(bilinear) [27]	70.0	63.7	71.6
NTN [17]	70.4	87.1	68.2
TransSparse(unif) [21]	86.8	86.5	87.4
TransSparse(bern) [21]	86.8	87.5	88.5
TransE (unif) [18]	75.9	70.9	77.3
TransE (bern) [18]	75.9	81.5	79.8
TransE-RS(unif) [29]	85.2	82.8	90.5
TransE-RS(bern) [29]	85.3	83.0	90.7
TransE-SS(unif)(this paper)	86.3	81.1	91.3
TransE-SS(bern)(this paper)	86.0	82.7	91.3
TransH (unif) [19]	77.7	76.5	74.2
TransH (bern) [19]	78.8	83.3	79.9
TransH-RS(unif) [29]	86.3	82.1	86.4
TransH-RS(bern) [29]	86.4	81.6	86.8
TransH-SS(unif)(this paper)	86.6	81.1	89.7
TransH-SS(bern)(this paper)	86.6	80.7	89.6
TransD (unif) [20]	85.6	85.9	86.4
TransD (bern) [20]	86.4	89.1	88.0
TransD-SS(unif)(this paper)	86.7	86.2	**92.3**
TransD-SS(bern)(this paper)	**86.8**	**89.2**	**92.3**

On FB15K, we train KG embeddings on training set, and use test set to test the scoring distributions. We adopt "unif" sampling method to generate negative triplets for both training and test sets.

For each pair of positive and negative triplets in test set, we compute three kinds of scores separately, the score $f_r(h,t)$ of positive triplet, the score $f_r(h',t')$ of negative triplet and the margin-score $f_r(h',t') - f_r(h,t)$ of the pair. For the test set, with 0.1 scoring interval, we count the proportion of triplets' scores in $(s - 0.05, s + 0.05]$ as the probability of score s.

Figs. 3a, 3b, 3c, 3d, 3e, and 3d show the results of TransE, TransH, TransE-RS, TransH-RS, TransE-SS and TransH-SS on the distributions of three kinds of scores, respectively. The score $f_r(h,t)$ distribution of the positive triplets is drawn by a solid line, the score $f_r(h',t')$ distribution of the negative triplets is drawn by a dash line, and the margin-score distribution are drawn by a dot line. The parameters for scoring functions are as follow. TransE: $\mu_1 = 1$; TransH: $\mu_1 = 0.25$; TransE-RS: $\mu_1 = 2$, $\mu_2 = 6$; TransH-RS: $\mu_1 = 2$, $\mu_2 = 6$; TransE-SS: $\varepsilon = 0$, $\mu_2 = 4$, $\mu_3 = 5$; TransH-SS: $\varepsilon = 0$, $\mu_2 = 7$, $\mu_3 = 8$.

According to these parameters, Trans(E,H) with L_R loss expect the marginal scores $f_r(h',t') - f_r(h,t) \geq \mu_1$, Trans(E,H)-RS with L_{RS} loss do not only try $f_r(h',t') - f_r(h,t) \geq \mu_1$, but also limit positive scores $f_r(h,t) \leq \mu_2$, and Trans(E,H)-SS with our L_{SS} loss can realize double limit scoring $f_r(h,t) \leq \mu_2$ and $f_r(h',t') \geq \mu_3$. In Figs. 3a and 3b, we can see the marginal scores of TransE and TransH are larger than 1 and 0.25 (dot lines) respectively. In Figs. 3c and 3d, for both TransE-RS and TransH-RS, the marginal scores are larger than 2 (dot lines) and the scores of positive

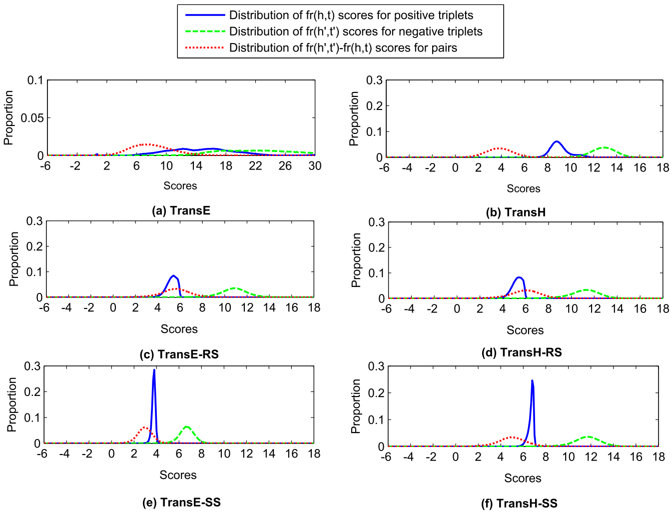


Fig. 3. Distribution of triplets on different scores(FB15K).

triplets are lower than 6 (solid lines). In Fig. 3e, TransE-SS takes $f_r(h,t) \leq 4$ (solid lines) and $f_r(h',t') \geq 5$ (dash lines), and in Fig. 3f, TransH-SS limits $f_r(h,t) \leq 7$ (solid lines) and $f_r(h',t') \geq 8$ (dash lines).

Comparing the scores' distributions of models by three kinds of loss functions, we can find an important difference in the positive triplets' score distributions. See distribution lines (solid lines) in Fig. 3, among the compared models our L_{SS} based models TransE-SS and TransH-SS have the most centralized proportion distribution on the scoring of positive triplets, and meanwhile have smaller overlaps between the positive and negative triplets' scores.

4.3.2 Performance on Different λ

In double scoring loss function L_{SS}, λ is used to weigh the lower-limit scoring loss for negative triplets, which is another important factor for our proposed models. This subsection mainly explores the availability of λ for the tasks of link prediction and triplet classification. In this experiment, we only conduct the tasks of link prediction and triplet classification on FB15k with different λ, and set λ from $\{0, 0.001, 0.1, 0.25, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. The rest optimal configuration also lies on Tables 6 and 7. Thus we only train models on training set with different λ, and then obtain the results of link prediction and triplet classification on test set.

The results of link prediction and triplet classification on FB15k is given by Fig. 4, which we can analyze that (1) For both tasks of link prediction and triplet classification, $\lambda = \{1, 2, 3\}$ is the best configuration and is suitable for TransE-SS, TransH-SS, TransD-SS and ProjE-SS, indicating that such setting best express semantic information of entity space and relation space. (2) The performance of our all extended models is first increasing with the growth of λ and then drops with λ further increases, which suggests that too lower or too higher λ damages the gradient balance of positive and negative triplets in training process.

4.3.3 Performance of Different μ_2 and μ_3

By considering the limit-based scoring loss item, the proposed double scoring loss L_{SS} is more effective compared

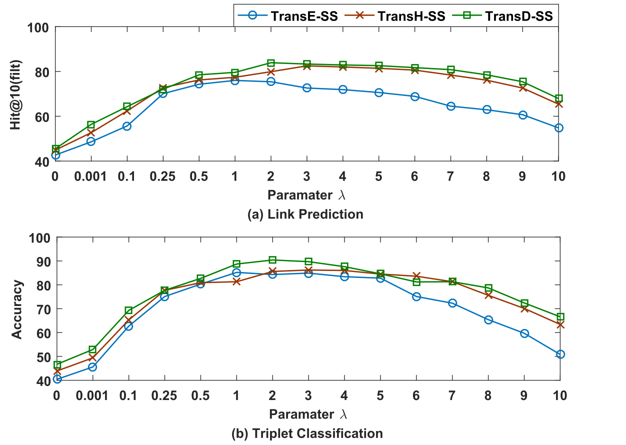


Fig. 4. Results with different λ on FB15K.

with the traditional margin-based ranking loss L_R and combined limit-based scoring loss L_{RS} for learning KG embeddings. In double scoring loss L_{SS}, μ_2 is the upper score margin for all positive triplets and μ_3 is the lower score margin for all negative triplets. Noted that $\mu_3 > \mu_2$ can guarantee the the margin between the scores of positive and negative triplets. Therefore we further analyse the correlativity of μ_2 and μ_3 on four datasets.

From the experimental results of translation-based models TransE-SS, TransH-SS and TransD-SS, we find that $\mu_3 = \{1, 2, 3, 4\} + \mu_2$ is available to WN18, FB13 and FB15k for link prediction and triplet classification, while there is the correlation of $\mu_3 = \{8, 9, 10\} + \mu_2$ to WN11 for triplet classification. For explanation, we give the results of all translation-based proposed models on four datasets under 'bern' setting in Fig. 5. In this experiment, we set different ratio of

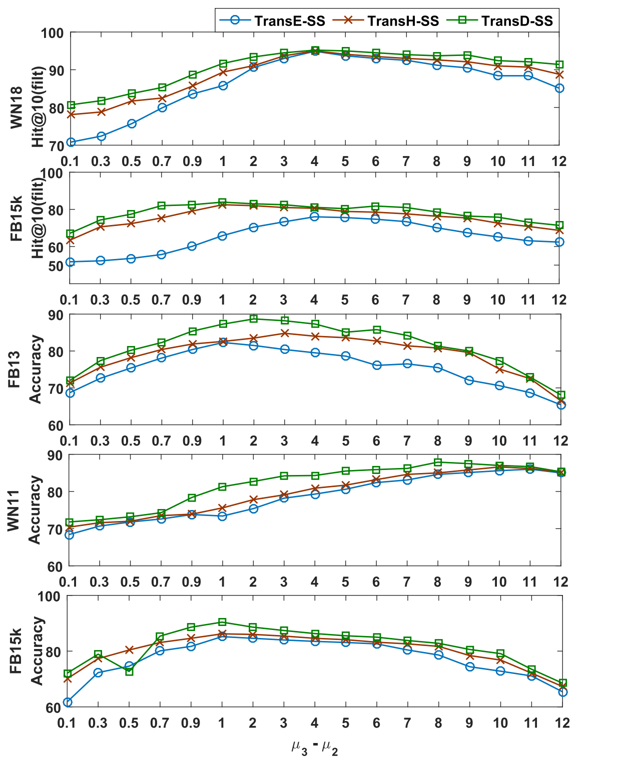


Fig. 5. The correlation of μ_2 and μ_3 for four datasets.

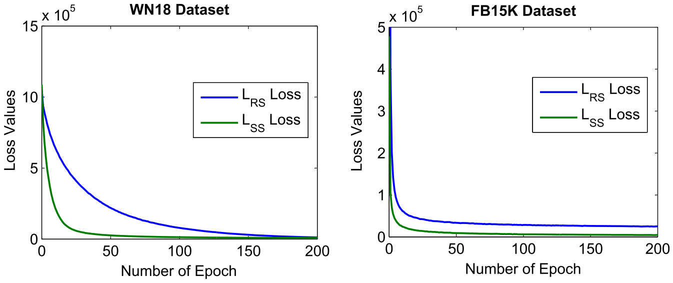


Fig. 6. Convergence of loss function.

μ_2 to μ_3 with fixed λ, that is $\mu_3 = \{0.1, 0.3, 0.5, 0.7, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\} + \mu_2$, and the rest optimal configurations are illustrated in Tables 6 and 7.

Complex relations such as "1-to-N", "N-to-1" and "N-to-N" relations exist in KGs, an entity may connects multiple relations simultaneously, and also needs to satisfy those relation golden conditions. It leads that the score of the entity's positive triplet is lower for one relation while maybe higher for other relations. Similarly, the score of an entity's negative triplets also has the above phenomenon. Therefore the lower μ_2 for limiting the upper scores of all positive triplets and higher μ_3 for limiting the lower scores for all negative triplets, are expected to make entities suit the evaluations for all the relations. Analysing from Fig 5, we find that too lower or too higher $\mu_3 - \mu_2$ are not suitable for learning embeddings of KGs, which can make an unstable and inconsistent results on four datasets for extended models. The setting $\mu_3 > \mu_2$ also maintains the characteristic of encouraging the discrimination between positive and negative triplets. Thus both μ_2 and μ_3 are important factors for our proposed models.

4.3.4 Convergence of Loss

Compared with the combined limit-based scoring loss L_{RS}, our double scoring loss L_{SS} considers the limitation on scores of all positive and negative triplets independently at the same time. Intuitively, our double scoring loss has a faster convergent rate than the combined limit-based scoring loss L_{RS}, since independent scoring limits for positive and negative triplets can provide more effective and flexible way of parameter learning in the training stage. In order to study how L_{RS} and L_{SS} behave converge, we test TransH-RS and TransH-SS on WN18 and FB15K datasets. The loss values of TransH-RS and TransH-SS with increasing epoch numbers are shown in Figure 6. We can see that the proposed L_{SS} loss can converge faster than L_{RS} loss [29], and also can reach persistent smaller loss values than the L_{RS} loss. So we can also conclude that L_{SS} loss with independent scoring limits for positive and negative triplets, provides faster parameter learning efficiency than L_{RS} loss.

5 Conclusions

In this paper, we propose a novel double scoring loss framework for learning KG embeddings. The key idea of our proposal double scoring loss is to independently enforce a lower upper-limit scoring on all positive and a higher lower-limit scoring on all negative triplets. Such two limitations guarantee the discrimination between positive and negative triplets. Moreover, upon the double scoring loss framework, we present several extended models TransE-SS, TransH-SS, TransD-SS, ProjE-SS and ComplEx-SS for KG embedding. We empirically conduct extensive experiments on triplet classification and link prediction with two KGs FreeBase and WordNet. The experimental results show that our double scoring loss has significantly and consistently considerable improvement over former corresponding KG embedding models, meanwhile has faster and more effective parameter learning rate.

Acknowledgments

We would like to thank all the reviewers, and deeply appreciate their valuable and constructive comments which significantly improve the quality of this paper. This work was supported by National Key R&D Program No.2016YFB0801300, by National Natural Science Foundation of China No.61202226 and No.11671379, and by U.S. National Science Foundation through Grant No.IIS-1763452.

References

[1] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, Jr., and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 1306–1313.

[2] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in *Proc. Int. Conf. Manage. Data SIGMOD*, 2012, pp. 481–492.

[3] J. Cheng, Z. Wang, J. R. Wen, J. Yan, and Z. Chen, "Contextual text understanding in distributional semantic space," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 133–142.

[4] M. Ashburner, C. A. Ball, and J. A. E. A. Blake, "Geneontology: Tool for the unification of biology," *Nat. Genet.*, vol. 25, no. 1, pp. 25–29, 2000.

[5] M. Nickel, V. Tresp, and H.-P. Kriegel, "Factorizing YAGO: scalable machine learning for linked data," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 271–280.

[6] A. Bordes, J. Weston, and N. Usunier, "Open question answering with weakly supervised embedding models," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2014, pp. 165–180.

[7] M. Yahya, K. Berberich, S. Elbassuoni, and G. Weikum, "Robust question answering over the web of linked data," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2013, pp. 1107–1116.

[8] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014b, pp. 615–620.

[9] S. He, K. Liu, Y. Zhang, L. Xu, and J. Zhao, "Question answering over linked data using first-order logic," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1092–1103.

[10] W. Shen, J. Wang, P. Luo, and M. Wang, "Linking named entities in tweets with knowledge base via user interest modeling," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 68–76.

[11] S. R. Szumlanski and F. Gomez, "Automatically acquiring a semantic network of related concepts," in *Proc. 19th ACM Conf. Inf. Knowl. Manage.*, 2010, pp. 19–28.

[12] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proc. Meet. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2011, pp. 541–550.

[13] L. Yao, S. Riedel, and A. Mccallum, "Probabilistic databases of universal schema," in *Proc. Joint Workshop Autom. Knowl. Base Construction Web-Scale Knowl. Extraction*, 2012, pp. 116–121.

[14] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proc. 54th Annu. Meet. Assoc. Comput. Linguistics*, 2016, pp. 2124–2133.

[15] Y. Jia, Y. Wang, H. Lin, X. Jin, and X. Cheng, "Locally adaptive translation for knowledge graph embedding," in *Proc. AAAI*, 2016, pp. 992–998.

[16] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proc. AAAI*, 2016, pp. 189–205.

[17] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *In Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 926–934.

[18] A. Bordes, N. Usunier, and A. Garcia-Duran, "Translating embeddings for modeling multi-relational data," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[19] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI*, 2014, pp. 1112–1119.

[20] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. ACL*, 2015, pp. 687–696.

[21] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 985–991.

[22] J. Feng, M. Huang, M. Wang, M. Zhou, Y. Hao, and X. Zhu, "Knowledge graph embedding by flexible translation," in *Proc. 15th Int. Conf. Princ. Knowl. Representation Reasoning*, 2016, pp. 557–560.

[23] H. G. Yoon, H. J. Song, S. B. Park, and S. Y. Park, "A translation-based knowledge graph embedding preserving logical property of relations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 907–916.

[24] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 301–306.

[25] I. Sutskever, J. B. Tenenbaum, and R. Salakhutdinov, "Modelling relational data using bayesian clustered tensor factorization," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1821–1828.

[26] R. Jenatton, N. L. Roux, A. Bordes, and G. R. Obozinski, "A latent factor model for highly multi-relational data," in *Proc. Int. Conf. Neural Inf. Process. Syst*, 2012, pp. 3167–3175.

[27] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, 2012, pp. 127–135.

[28] A. Bordes, J. Weston, X. Glorot, and Y. Bengio, "A semantic matching energy function for learning with multirelational data," *Mach. Learn.*, vol. 94, no. 2, pp. 233–259, 2014.

[29] X. Zhou, Q. Zhu, P. Liu, and L. Guo, "Learning knowledge embeddings by combing limit-based scoring loss," in *Proc. 26th ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1009–1018.

[30] B. Shi and T. Weninger, "ProjE: Embedding projection for knowledge graph completion," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1236–1242.

[31] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 2071–2080.

[32] G. A. Miller, "WordNet: A lexical database for english." *Commun. ACM*, vol. 38, no. 11, pp. 39–40, 1995.

[33] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase:a collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1247–1250.

[34] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2181–2187.

[35] D. Q. Nguyen, K. Sirts, L. Qu, and M. Johnson, "StransE: A novel embedding model of entities and relationships in knowledge bases," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 460–466.

[36] Q. Xie, X. Ma, Z. Dai, and E. Hovy, "An interpretable knowledge transfer model for knowledge base completion," in *Proc. Meet. Assoc. Comput. Linguistics*, 2017, pp. 950–962.

[37] M. Nickel, V. Tresp, and H. P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 809–816.

[38] R. A. Harshman and M. E. Lundy, "PARAFAC: Parallel factor analysis," *Comput. Statist. Data Anal.*, vol. 18, no. 1, pp. 39–72, Aug. 1994.

[39] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[40] R. A. Harshman, "Models for analysis of asymmetrical relationships among n objects or stimuli," *Psychometrika*, vol. 31, no. 3, 1966, Art. no. 279.

[41] T. G. Kolda, B. W. Bader, and J. P. Kenny, "Higher-order web link analysis using multilinear algebra," in *Proc. IEEE Int. Conf. Data Mining*, 2005, pp. 242–249.

[42] T. Franz, A. Schultz, S. Sizov, and S. Staab, "TripleRank: Ranking semantic web data by tensor decomposition," in *Proc. Int. Semantic Web Conf.*, 2009, pp. 213–228.

[43] L. Drumond, S. Rendle, and L. Schmidt-Thieme, "Predicting RDF triples in incomplete knowledge bases with tensor factorization," in *Proc. ACM Symp. Appl. Comput.*, 2012, pp. 326–331.

[44] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2010, pp. 81–90.

[45] S. Rendle, "Scaling factorization machines to relational data," in *Proc. Int. Conf. Very Large Data Bases*, 2013, pp. 337–348.

[46] P. Miettinen, "Boolean tensor factorizations," in *Proc. IEEE Int. Conf. Data Mining*, 2012, pp. 447–456.

[47] B. Yang, W. T. Yih, X. He, J. Gao, and L. Deng, "Learning multi-relational semantics using neural-embedding models," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–7.

[48] N. Maximilian, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1955–1961.

[49] T. Dettmers *et al.*, "Convolutional 2d knowledge graph embeddings," *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, pp. 1811–1818, 2018.

[50] Y. Lin, Z. Liu, and M. Sun, "Modeling relation paths for representation learning of knowledge bases," in *Proc. Empir. Methods Natural Lang. Process.*, 2015, pp. 705–714.

[51] A. Garcia-Duran, A. Bordes, and N. Usunier, "Composing relationships with translations," in *Proc. Empir. Methods in Natural Lang. Process.*, 2015, pp. 286–290.

[52] Z. Wei, J. Zhao, and K. Liu, "Mining inference formulas by goal-directed random walks," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 1379–1388.

[53] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–18.

[54] T. Trouillon and M. Nickel, "Complex and holographic embeddings of knowledge graphs: A comparison," in *Proc. Int. Workshop Statist. Relational AI, 2017*, 2017, pp. 1–6.

[55] D. Ruffinelli, S. Broscheit, and R. Gemulla, "You can teach an old dog new tricks!on training knowledge graph embeddings," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–20.

[56] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, 2010, pp. 177–186.

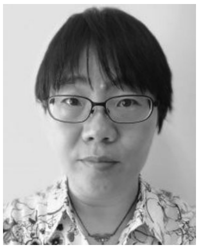
[57] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *Proc. Empir. Methods Natural Lang. Process.*, 2015, pp. 1499–1509.

Xiaofei Zhou received the PhD degree in pattern recognition and artificial intelligence from the Nanjing University of Science and Technology, China, in 2008. She as a postdoctor worked in the Graduate University of Chinese Academy of Sciences from 2008 to 2010. She is currently a professor with the Institute of Information Engineering, Chinese Academy of Sciences, and also a teaching professor at UCAS. Her research interests include machine learning, natural language processing, and pattern recognition.

Lingfeng Niu received the BS degree in mathematics from Xian Jiaotong University, in 2004, and the PhD degree in mathematics from the Chinese academy of sciences, in 2009. She has been a professor with the Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, since 2009. Her current research interests include optimization and data mining.

Qiannan Zhu received the PhD degree from the Institute of Information Engineering, University of Chinese Academy of Sciences, in 2020. Her current research interests include data mining, knowledge graph representation, deep learning, and natural language process.

Xingquan Zhu (Senior Member, IEEE) received the PhD degree in computer science from Fudan University, China. He is currently a professor with the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA. His research interests include data mining, machine learning, and multimedia systems. Since 2000, he has authored or co-authored more than 250 refereed journal and conference papers in these areas, including three best paper awards and one Best Student Award. He is an associate editor of *IEEE Transactions on Knowledge and Data Engineering* (2008–2012, and 2014–date), and an associate editor of *ACM Transactions on Knowledge Discovery from Data* (2017–date).



Ping Liu is currently an associate professor with the Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include data mining and pattern matching.

Jianlong Tan received the PhD degree from the institute of Computing Technology, Chinese Academy of Sciences. He is currently a professor with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include data stream management systems and data mining.

Li Guo is a professor with the Institute of Information Engineering, Chinese Academy of Sciences. She is also the director in the Intelligent Information Processing Research Center, Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include data stream management systems and data mining.