Knowledge Graph Embedding by Adaptive Limit Scoring Loss Using Dynamic Weighting Strategy

Jinfa Yang, Xianghua Ying*, Yongjie Shi, Xin Tong, Ruibin Wang, Taiyan Chen, Bowei Xing
Key Laboratory of Machine Perception (MOE)
School of Artificial Intelligence, Peking University
{jinfayang, xhying, shiyongjie, xin_tong, robin_wang}@pku.edu.cn, chenty@stu.pku.edu.cn, 2017xbw@pku.edu.cn

Abstract

Knowledge graph embedding aims to represent entities and relations as low-dimensional vectors, which is an effective way for predicting missing links in knowledge graphs. Designing a strong and effective loss framework is essential for knowledge graph embedding models to distinguish between correct and incorrect triplets. The classic margin-based ranking loss limits the scores of positive and negative triplets to have a suitable margin. The recently proposed Limit-based Scoring Loss independently limits the range of positive and negative triplet scores. However, these loss frameworks use equal or fixed penalty terms to reduce the scores of positive and negative sample pairs, which is inflexible in optimization. Our intuition is that if a triplet score deviates far from the optimum, it should be emphasized. To this end, we propose Adaptive Limit Scoring Loss, which simply re-weights each triplet to highlight the less-optimized triplet scores. We apply this loss framework to several knowledge graph embedding models such as TransE, TransH and ComplEx. The experimental results on link prediction and triplet classification show that our proposed method has achieved performance on par with the state of the art.

1 Introduction

Knowledge graphs are usually collections of factual triplets — (head entity, relation, tail entity), also known as (subject, predicate, object), which represent human knowledge of the real world in a structured way. There are some outstanding knowledge graphs, such as WordNet (Miller, 1995), Freebase (Bollacker et al., 2008), DBpedia (Lehmann et al., 2015), YAGO (Suchanek et al., 2007). They have gained widespread attention for their successful usage in various applications, e.g., question answering (Bordes et al., 2014; Huang et al., 2019).

*Corresponding Author

Figure 1: Comparison between the popular optimization manner of reducing $(S_n, S_p)$ and the proposed reducing $(\alpha_n S_n, \alpha_p S_p)$. (a) Reducing $(S_n, S_p)$ is prone to inflexible optimization ($P_1, P_2$ and $P_3$ all have equal gradients with respect to $S_n$ and $S_p$), as well as potential overlapping problem (both $T$ and $T'$ on the decision boundary are acceptable). (b) With $(\alpha_n S_n, \alpha_p S_p)$, the $L_{AS}$ dynamically adjusts its gradients on $S_p$ and $S_n$, and thus benefits from a flexible optimization process. For $P_1$, it emphasizes on increasing $S_n$; for $P_3$, it emphasizes on reducing $S_p$. Moreover, it aggregates $T$ and $T'$ on the circular decision boundary, which can alleviate the overlap problem.

recommendation systems (Zhou et al., 2020), medical science (Hasan et al., 2020), etc.

Similar to word embedding, knowledge graph embedding is one of the basic research fields of knowledge graph, which can be applied to tasks such as knowledge graph completion (Bordes et al., 2013; Sun et al., 2019), triplet classification (Socher et al., 2013; Nguyen et al., 2020), search personalization (Lu et al., 2020). For a knowledge graph embedding model, there are two major components, the scoring triplets and the optimizing loss function. In the last few years, negative sampling with margin-based ranking loss framework has been commonly used for modelling knowledge graph embedding. In this framework, a positive triplet $(h, r, t)$ can get its score $S_p = f_r(h, t)$, and the corresponding negative triplet $(h', r, t')$ score value is $S_n = f_r(h', t')$, where $f_r$ is the scoring function. Finally, optimize the margin-based
ranking loss function \( \max(0, \mu + S_p - S_n) \). In \( \max(0, \mu + S_p - S_n) \), increasing \( S_p \) is equivalent to reducing \( S_n \). We argue that this symmetric optimization manner is prone to the following two problems.

**Lack of flexibility in optimization.** The penalty strength on \( S_p \) and \( S_n \) is restricted to be equal or fixed. Given the specified loss function, the gradients of \( S_p \) and \( S_n \) have the same amplitude or fixed multiples. In some corner cases, e.g., when both \( S_p \) and \( S_n \) are small ("P1" in Figure 1a), we expect positive samples \( S_p \) to be small and negative samples \( S_n \) to be large, so we need a smaller penalty for \( S_p \) and a larger penalty for \( S_n \). However, the aforementioned loss framework also retains a large gradient magnitude for \( S_p \), which is inefficient and irrational.

**Overlapping between \( S_p \) and \( S_n \).** Under a margin-based ranking loss (exclude \( \{S_{p_1}^h, S_{p_1}^r\} \) here), there are three kinds of value distributions for a pair of positive and negative triplets \( \{(h, t), (h', t')\} \), including \( \{S_{p_1}^h, S_{h_1}^r\}, \{S_{p_1}^l, S_{l_1}^r\}, \{S_{p_2}^h, S_{h_2}^r\} \) where the superscript \( i \) indicates a low value, \( h \) indicates a high value, and the number indicates three cases. As long as \( S_{p_1}^i - S_{h_1}^i < -\mu \), \( i = 1, 2, 3 \) is satisfied, there may be an overlap phenomenon of \( S_{p_2}^h > S_{l_1}^l \). For example, \( T \) (one of the optimized states) has \( \{S_p, S_n\} = \{1, 4\} \) and \( T' \) has \( \{S_{p_1}^l, S_{l_1}^r\} = \{5, 8\} \). They are both satisfied with the margin of \( \mu = 3 \). However, when comparing them against each other, we find \( S_{p_2}^h > S_{l_1}^l \). The overlap between \( S_p \) and \( S_n \) damages the separability of positive and negative triplets.

Limit-based scoring loss (Zhou et al., 2017) proposes to add an upper-limit scoring loss on \( f_c(h, t) \) to guarantee low scores for the positive triplets, which can effectively avoid \( \{S_{p_2}^h, S_{h_2}^r\} \) case; Double limit scoring loss (Zhou et al., 2021) adds a lower-limit score for negative triplets on this basis, and finally alleviates the overlap problem. However, neither method can solve the problem of inflexible optimization. Our intuition is that if a triplet score deviates far from the optimum, it should be emphasized. To this end, we propose Adaptive Limit Scoring Loss, which simply reweights each triplet to highlight the less-optimized triplet scores. The main contributions of this paper are summarized as follows:

- We propose adaptive limit scoring loss, which benefits knowledge graph embedding with flexible optimization and definite positive and negative triplet separation.
- Compared with the recent knowledge graph embedding negative sample loss framework limit-based scoring loss and double limit scoring loss (Zhou et al., 2017, 2021), our method not only reduces the amount of tuning parameters but also improves the performances.
- Experiments are carried out on WordNet and Freebase datasets with link prediction and triplet classification task, and the results show the superiority of our proposed method with performance on par with the state of the art.

2 Related Works

2.1 Knowledge Graph Embedding Models

Roughly speaking, we can divide knowledge graph embedding models into translational distance models and semantic matching models.

**Translational distance models** describe relations as translations from source entities to target entities. TransE (Bordes et al., 2013) is the most widely used translation distance constraint model. It assumes that entities and relations satisfy \( h + r = t \), where \( h, r, t \in \mathbb{R}^k \). However, TransE cannot handle 1-N, N-1, and N-N relations well (Wang et al., 2014). TransH (Wang et al., 2014) is proposed to compensate for the shortcomings of TransE. It projects entities onto relationspecific hyperplanes with \( h_\perp = h - w^h_r h_w \), and \( t_\perp = t - w^r_t t_w \). TransR (Lin et al., 2015) has a very similar idea to TransH, which introduces relation-specific spatial transformations instead of hyperplanes. TransE_AT (Yang et al., 2021) improves TransE’s ability to express symmetric relations by introducing affine transformation. TranSparse (Ji et al., 2016) simplifies TransR by forcing the projection matrix to be sparse. Moreover, RotatE (Sun et al., 2019) defines each relation as a rotation from the source entity to the target entity in a complex vector space, which can represent various relation patterns including symmetry/asymmetry, inversion and composition.

**Semantic matching models** use the similarity scoring function to evaluate the latent semantics of entities and relations. RESCAL (Nickel et al., 2011) is a tensor factorization model which represents each relation as a full-rank matrix and defines score function as \( f_c(h, t) = (h^\top M_r t) \). DistMult (Yang et al., 2015) simplifies the embedding of relations \( M_r \) as a diagonal matrix, which
can reduce the number of parameters and make the model easier to train. However, Distmult assumes that all relations are symmetric, and is not friendly to other types of relations, such as anti-symmetry and composition. To solve this problem, ComplEx (Trouillon et al., 2016) extends DistMult to complex space: \( \mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^d \), and uses conjugate-transpose \( \mathbf{t} \) to model asymmetric relations. MLP (Dong et al., 2014) and NTN (Socher et al., 2013) use a fully connected neural network to calculate the scores of given triplets. TransE (Bordes et al., 2013), TransH (Wang et al., 2014), TransR (Lin et al., 2015), etc. have successfully been used for NTN (Socher et al., 2013), TransE (Bordes et al., 2013), TransH (Wang et al., 2014), TransR (Lin et al., 2015), etc. The loss framework has been proved to be successfully applied to knowledge graph embedding in recent years. In this paper, our framework mainly focuses on improving the marginal discrimination between positive and negative triplets’ scores but also low scores for positive triplets and high scores for negative triplets.

### 2.2 Loss Functions

For knowledge graph embedding models optimized with negative sampling, we summarize the related loss functions as follows.

Margin-based ranking loss \( L_R \) is a widely used loss function for KG embedding models, which has successfully been used for NTN (Socher et al., 2013), TransE (Bordes et al., 2013), TransH (Wang et al., 2014), TransR (Lin et al., 2015), etc. The \( L_R \) is formulated by:

\[
L_R = \sum_{(h,r,t) \in \mathcal{G}} [\mu + S_p - S_n]_+ , \tag{1}
\]

where \([x]_+ = \max(0,x)\) is a rectified linear unit that denotes the positive part of \( x \). \( \mu \) is the margin between positive and negative triplets, \( S_p = f_r(h,t), S_n = f_r(h',t') \) represents the score of the positive and negative triplets respectively. \( \mathcal{G} \) denotes the set of positive triplets, and \( \mathcal{G}' = \{(h',r,t) \notin \mathcal{G}, h' \in \mathcal{E}\} \cup \{(h,r,t') \notin \mathcal{G}, t' \in \mathcal{E}\} \) denotes the set of corrupted triplets.

Limit-based scoring loss (Zhou et al., 2017) adds an upper-limit scoring loss term \([S_p - \mu_p]_+\) to guarantee low scores for positive triplets. The loss framework has been proved to be successfully applied in TransE and TransH, and its formula is:

\[
L_{RS} = \sum_{(h,r,t) \in \mathcal{G}} [\mu + S_p - S_n]_+ + \lambda [S_p - \mu_p]_+ , \tag{2}
\]

where \( \lambda, \mu_p > 0 \). On this basis, Double Limit Scoring Loss (Zhou et al., 2021) proposes to replace \([\mu + S_p - S_n]_+\) of \( L_{RS} \) with lower-limit scoring loss for negative triplets \([\mu_n - S_n]_+\). The loss framework is:

\[
L_{SS} = \sum_{(h,r,t) \in \mathcal{G}} [S_p - \mu_p]_+ + \lambda [\mu_n - S_n]_+ , \tag{3}
\]

where \( \mu_n > \mu_p > 0 \). Compared with \( L_R \) and \( L_{RS} \) losses, \( L_{SS} \) loss expects not only marginal discrimination between positive and negative triplets’ scores but also low scores for positive triplets and high scores for negative triplets.

Some other negative sampling losses of the knowledge graph embedding model also try to improve the discrimination between positive and negative triplets. HoE (Nickel et al., 2016) suggests to use logistic function instead of rectified linear unit to distinguish the probabilities of positive and negative triplets. ComplEx (Trouillon et al., 2016) propose a negative log-likelihood loss to learn compact representations. ProjE (Shi and Weninger, 2017) uses the pointwise ranking method to optimize the list of candidate entities collectively, so that the probability ranking of positive triplets is higher than that of negative triplets. RotatE (Sun et al., 2019) defines a log-sigmoid function to make the positive and negative triplets away from the same margin in the opposite direction. Sun et al. (Sun et al., 2020) propose the pair similarity optimization and successfully apply the method in visual tasks such as face recognition. Inspired by this, we refine the scoring and weighting strategies and apply them to knowledge graph embedding. Except for negative sampling methods, neural network frameworks with cross-entropy loss (Lacroix et al., 2018) and 1-N binary cross-entropy loss (Dettmers et al., 2018) have been developed for knowledge graph embedding in recent years. In this paper, our work mainly focuses on improving the marginal ranking loss \( L_R \) and the limited loss \( L_{RS} & L_{SS} \) for knowledge graph embedding.

### 3 The Proposed Methods

In this section, we firstly present adaptive limit scoring loss \( L_{AS} \) for optimizing Knowledge graph embedding models. Secondly, we introduce different metrics of our loss for optimization according to the positioning method of the circle center.

#### 3.1 Adaptive Limit Scoring Loss

We consider enhancing the optimization flexibility by allowing each triplet score to learn at its
own pace, depending on its current optimization
status. Then, we add adaptive penalty items to the
positive and negative triplets scoring respectively. Equation (3) can be changed to:

\[ L_{AS} = \sum_{(h,r,t) \in \mathcal{G}} \alpha_p [S_p - \mu_p]_+ + \alpha_n [\mu_n - S_n]_+ . \]

(4)

Where \( \alpha_n \) and \( \alpha_p \) are non-negative weighting fac-
tors. During training, when back propagating to \( S_p \)
(\( S_n \)), the gradient with respect to \( \alpha_p [S_p - \mu_p]_+ + \alpha_n [\mu_n - S_n]_+ \) will be multiplied by \( \alpha_p(\alpha_n) \). When
the triplet score deviates far from its optimum (i.e.,
\( v_p \) for \( S_p \) and \( v_n \) for \( S_n \). \( v_p \) and \( v_n \) are intermediate
variables), it should obtain a large weighting factor in order to obtain effective update with large
gradient. To this end, we define \( \alpha_p \) and \( \alpha_n \) in an
adaptive way:

\[
\begin{aligned}
\alpha_p &= [S_p - v_p]_+ \\
\alpha_n &= [v_n - S_n]_+ ,
\end{aligned}
\]

(5)

Overall, the adaptive limit scoring loss in Equation (4) expects \( S_p < \mu_p \) and \( S_n > \mu_n \). We
further analyse the settings of \( \mu_p \) and \( \mu_n \) by de-

ering the decision boundary. In the optimization
process, the decision boundary is realized at
\( \alpha_p(S_p - \mu_p) + \alpha_n(\mu_n - S_n) = 0 \). Combined with
Equation (5), we can get:

\[
(S_p - \frac{v_p + \mu_p}{2})^2 + (S_n - \frac{v_n + \mu_n}{2})^2 = C ,
\]

(6)

where \( C = \left( (v_p - \mu_p)^2 + (v_n - \mu_n)^2 \right)/4 \). Equation (6) shows that the decision boundary is the arc
of a circle, as shown in Figure 1b. The center of the
circle is at \( S_n = (v_n + \mu_n)/2 \), \( S_p = (v_p + \mu_p)/2 \),
and the radius equals \( \sqrt{C} \). Here we have four hyper-
parameters \( \mu_p \) and \( \mu_n \) from Equation (4), \( v_p \) and \( v_n \) from Equation (5). After Positioning the
center of the circle, the four hyperparameters can be
reduced to two, which is less than \( L_{RS} \) and \( L_{SS} \).

3.2 Positioning the Center of Circle

The center of circle is the ideal optimization target
for \( (S_n, S_p) \), and the arc is the actual decision
boundary. Usually, we expect lower score for \( S_n \)
and higher for \( S_p \). However, our model training is
based on the open world assumption, which states
that knowledge graphs contain only true facts and
non-observed facts can be either false or just miss-
ing (Drumond et al., 2012). It means that the gen-
erated negative triplets may be correct, but they do
not appear in the original knowledge graph. There-
fore, we do not want \( S_n \) to be infinite but a finite
value. Here we consider two options:

**Constant Adaptive Limit Scoring Loss (CAS).**

We set the center of the circle as a constant \((0, \mu_p + \mu_n)\). Correspondingly, the two hyper-param-
ers \( v_p \), \( v_n \) in Equation (5) can be reduced by setting
\( v_p = -\mu_p \), \( v_n = \mu_n + 2\mu_p \). And the decision
boundary in Equation (6) can be degraded into:

\[
(S_p - 0)^2 + (S_n - (\mu_p + \mu_n))^2 = 2\mu_p^2 .
\]

(7)

The decision boundary defined in Equation (7)
aims to optimize \( S_p \rightarrow 0 \) and \( S_n \rightarrow \mu_p + \mu_n \) (Actu-
ally \( 0, \mu_p + \mu_n \) cannot be reached, in Equation (4)
we limit \( S_p > \mu_p, S_n < \mu_n \)). The choice of the constant
\((\mu_p + \mu_n)\) is inspired by the value range of the
dynamic weighting in Equation (5). When
the model embedding needs to be optimized (that
is, \( S_p > \mu_p, S_n < \mu_n \)), substituting \( v_p = -\mu_p \)
into Equation (5), we can get the positive triplet
dynamic weight range \( \alpha_p > 2\mu_p \). Similarly, substi-
tuting \( v_n = \mu_n + 2\mu_p \) into Equation (5), we can get
the same range of negative triplets dynamic weight
\( \alpha_n > 2\mu_p \).

**Independent Adaptive Limit Scoring Loss (IAS).**

When the model embedding is in different states (such as \( P_1 \), \( P_2 \) and \( P_3 \) in Figure 2), it should have
different optimized trajectories. We expect
to find the optimal trajectory for each independent
embedding state. Taking point \( P_1 \) (assume its coor-
dinates are \( (S_n, S_p) \)) in Figure 2 as an example, its
corresponding decision boundary is the largest arc
(located in light blue sector), and the center of the

Figure 2: Different embedding states have different opti-
mization trajectories. \( P_1, P_2, \) and \( P_3 \) have different ideal
optimization goals and derive three decision boundary
arcs (located in light blue, green and red sectors).
circle is \( P_{C1}(C_{1n}, 0) \). Based on triangle similarity
\[ \Delta P_{C1}P_0P_1' \sim \Delta P_{C1}P_1P_0' \]
we can get:
\[ C_{1n} = \mu_n + \frac{\mu_n - S_n}{S_p - \mu_p}, \quad (8) \]
where \( S_n < \mu_n, S_p > \mu_p \). Combing the center of circle defined by Equation (6), the two hyper-parameters \( v_p, v_n \) in Equation (5) can be reduced by setting \( v_p = -\mu_p, v_n = \mu_n + 2\mu_p (\mu_n - S_n)/(S_p - \mu_p) \). Compared with \( L_{CAS}, L_{IAS} \) can independently set the circle center of each sample to obtain an independent optimized trajectory.

Adaptive Limit Scoring \( L_{AS} \) further improves double scoring loss \( L_{SS} \) by adding adaptive penalty terms to dynamically adjust the optimization process. In the early stage of model training, the scores of the positive and negative triplets are far from optimization, which increases the weight of the penalty item and obtains a larger gradient. This is conducive to the early rapid convergence for the model. During training, when there is a bias in the optimization of the paired positive and negative triplets, e.g., the positive triplet is close to the optimum while the negative triplet is still far from the requirement, the penalty term will increase the weight of the negative triplet so that the negative triplet can be adjusted in time. In addition to the separate limits for the positive and negative scores, the differentiated pace adjustment with penalty items can also alleviate the overlap problem (see \( T' \) in Figure 1 a and b).

4 Experiments

We comprehensively evaluate the effectiveness of Adaptive Limit Scoring Loss for link prediction (Bordes et al., 2013) and triplet classification (Socher et al., 2013) tasks under different knowledge graph embedding models. Our experiments are carried out on two popular knowledge graphs FreeBase (Bollacker et al., 2008) and WordNet (Miller, 1995). Freebase contains a large number of world facts such as movies, sports. WordNet is a large-scale lexical knowledge graph. Some subsets of the two knowledge graphs are used in our experiments, including WN18, WN18RR and WN11 from WordNet, and FB15k, FB15K-237 and FB13 from Freebase. The statistics of these subsets are shown in Table 1. FB15K-237 (Toutanova and Chen, 2015) and WN18RR (Dettmers et al., 2018) are subsets of FB15k and WN18, respectively, where inverse relations are deleted.

| Dataset    | #En | #Re | #train | #valid | #test |
|------------|-----|-----|--------|--------|-------|
| WN18       | 40,943 | 18 | 141,442 | 5,000  | 5,000 |
| FB15K      | 14,951 | 1,345 | 483,142 | 50,000 | 59,071 |
| WN18RR     | 40,943 | 11 | 86,835  | 3,034  | 3,134 |
| FB15k-237  | 14,541 | 237 | 272,115 | 17,535 | 20,466 |
| WN11       | 38,696 | 11 | 112,581 | 2,609  | 10,544 |
| FB13       | 75,043 | 13 | 316,232 | 5,908  | 23,733 |

Table 1: Number of entities, relations, and observed triplets in each split for benchmarks.

Parameters Settings. We compare the series of TransE, TransH, RotatE and ComplEx with different losses. The ranges of the main hyperparameters for the grid search are set as follows: learning rate \( \alpha \in \{0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01\} \), the embedding dimension \( m \in \{50, 80, 100, 150, 200\} \), the batch size \( B \in \{50, 100, 200, 500, 1000, 2000, 5000\} \), \{L1, L2\} distances for loss functions. For TransE and TransH with Adaptive Limit Scoring, upper limit score for positive triplets \( \mu_p \in \{0.25, 1, 2, 3, 4, 5, 6, 7, 8, 10, 15\} \), and lower limit score for negative triplet \( \mu_n \in \{0.1, 0.25, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\} \). Parameter \( C \) for TransH series from \{0.0005, 0.0625, 0.25, 1.0\}. For ComplEx, upper limit \( \mu_p \) score for positive triplets is \( log(p_+) \), \( p_+ \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\} \), and lower limit score \( \mu_n \) for negative triplet \( log(p_-) \), \( p_- \in \{p_+ \cdot \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}\) \}. We train WN18 and FB15K with 1000 times, WN18RR and FB15K237 with 3000 times for Link prediction, WN11, FB13 and FB15K with 1000 times for triplet classification. For RotatE, we use the parameters recommended by Sun et al. (2019) (with larger epoch, embedding dim and self-adversarial negative sampling) and the same \( \mu_p, \mu_n \) parameter search range as TransE and TransH. We use SGD for TransE, TransH and Adam (Kingma and Ba, 2014) for RotatE, ComplEx as the optimizer and fine-tune the hyperparameters on the validation dataset.

4.1 Link Prediction

Link prediction (Bordes et al., 2012, 2013) aims to predict the missing triplets such as head entity prediction \((?, r, t)\) or tail entity prediction \((h, r, ?)\) based on the known triplets. For a testing triplet \((h, r, t)\), either the head entity \( h \) or the tail entity \( t \) will be replaced with the total list of the embedding entities to construct the predicted triplets. Then
| Models                | WN18 Mean Hits@10(%) | FB15k Mean Hits@10(%) |
|----------------------|----------------------|-----------------------|
|                      | raw filt             | raw filt              | raw filt |
| RESCAL               | 1,180 1,163          | 37.2 52.8             | 828 683 28.4 44.1 |
| SME(linear)          | 545 533              | 65.1 74.1             | 274 154 30.7 40.8 |
| SME(bilinear)        | 526 509              | 54.7 61.3             | 284 158 31.3 41.3 |
| TransR(unif)         | 232 219              | 78.3 91.7             | 226 78 43.8 65.5 |
| TransR(bern)         | 238 225              | 79.8 92.0             | 198 77 48.2 68.7 |
| TransSparse(unif)    | 233 221              | 79.6 93.4             | 216 66 50.3 78.4 |
| TransSparse(bern)    | 223 211              | 80.1 93.2             | 190 82 53.7 79.9 |
| DistMult             | 987 902              | 79.2 93.6             | 224 97 51.8 82.4 |
| STransE              | 217 206              | 80.9 93.4             | 219 69 51.6 79.7 |
| TransE(unif)         | 263 251              | 75.4 89.2             | 243 125 34.9 47.1 |
| TransE-RS(unif)      | 362 348              | 80.3 93.7             | 161 62 53.1 72.3 |
| TransE-RS(bern)      | 385 371              | 80.4 93.7             | 161 65 53.2 72.1 |
| TransE-S(unif)       | 285 279              | 83.1 94.4             | 170 39 54.3 78.7 |
| TransE-SS(unif)      | 276 263              | 83.6 95.0             | 155 54 55.8 76.5 |
| TransE-CAS(unif)(ours) | 164 153          | 83.0 95.2             | 178 55 54.8 83.3 |
| TransE-CAS(bern)(ours) | 163 153         | 83.1 95.3             | 160 54 55.8 81.4 |
| TransE-IAS(unif)(ours) | 182 172          | 83.4 95.3             | 174 46 55.4 85.1 |
| TransE-IAS(bern)(ours) | 176 166          | 83.5 95.4             | 155 50 56.2 81.6 |
| TransH(unif)         | 318 303              | 75.4 86.7             | 211 84 42.5 58.5 |
| TransH(bern)         | 401 388              | 73.0 82.3             | 212 87 45.7 64.4 |
| TransH-RS(unif)      | 401 389              | 81.2 94.7             | 163 64 53.4 72.6 |
| TransH-RS(bern)      | 371 357              | 80.3 94.5             | 178 77 53.6 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 |
| TransH-CAS(unif)(ours) | 209 196          | 83.6 95.1             | 215 58 54.1 83.7 |
| TransH-CAS(bern)(ours) | 203 194          | 84.1 95.2             | 165 53 55.1 83.2 |
| TransH-IAS(unif)(ours) | 186 175          | 83.1 95.1             | 178 51 54.9 85.1 |
| TransH-IAS(bern)(ours) | 195 186          | 83.8 95.4             | 156 49 56.0 83.1 |
| ComplEx              | - -                 | - -                   | - - 84.0 |
| ComplEx-SS           | 431 418              | 84.0 95.9             | 179 53 53.8 85.9 |
| ComplEx-CAS(ours)    | 445 434              | 85.2 95.9             | 184 72 54.7 86.6 |
| ComplEx-IAS(ours)    | 441 432              | 84.3 95.8             | 197 83 54.6 85.9 |

Table 2: Evaluation results on WN18 and FB15k datasets. In each column, the top-1 result with bold marker and top-2-4 results with underline markers are given.

such triplets are ranked in descending order according to the scoring function. Based on the score rank, several metrics are usually reported: mean rank (MR), Mean Reciprocal Rank (MRR) and the proportion of top-k rank (Hits@k) for correct entities. A good model should have low “MR”, high “MRR” and high “Hits@k". For constructing the corrupted triplets, "unif" means that the head or tail entity is replaced with equal probability traditionally, and “bern” denotes reducing false negative labels by replacing head or tail with different probabilities (Wang et al., 2014). The settings “raw” and “filt” for the metrics distinguish whether or not to consider the impact of a corrupted triplet existing in the correct Knowledge graph.

4.1.1 Results on WN18 and FB15K

Firstly, we follow the experimental procedures of most negative sampling knowledge graph embedding models (such as Bordes et al. (2013); Wang et al. (2014), etc.), and use MR and Hits@10 to evaluate WN18 and FB15K. The optimal configurations are illustrated in Appendix A Table 5.

Table 2 shows the evaluation results on two datasets WN18 and FB15K. The original results of TransE, TransH and ComplEx are from the references (Bordes et al., 2013; Wang et al., 2014; Trouillon et al., 2016). And their extension with limit-based scoring loss (-RS), double limit scoring loss (-SS) are from Zhou et al. (2017, 2021) For the other compared models, we report the original results from Lin et al. (2015); Ji et al. (2016); Yang et al. (2014); Nguyen et al. (2016).

From Table 2, we can see that models with $L_{AS}$ (Including CAS and IAS refer to Section 3.2) loss have improved in different degrees. Compared to WN18 (95% + on hit@10) whose results are already high, FB15K has been improved significantly. On FB15K, the results (Compare in the best results for Hit@10) are increased by TransE 6.4%,
### 4.1.2 Results on WN18RR and FB15K-237

FB15K-237 (Toutanova and Chen, 2015) and WN18RR (Dettmers et al., 2018) are two more challenging datasets for Knowledge graph completions, where the inverse relations are deleted and the main relation patterns are symmetry/antisymmetry and composition patterns. In recent years, many embedding models (Dettmers et al., 2018; Sun et al., 2019) 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 loss framework on TransE, TransH, ComplEx and RotatE with their former loss models Zhou et al. (2017, 2021); Bordes et al. (2013); Wang et al. (2014); Trouillon et al. (2016); Sun et al. (2019) and some baseline models Rescal (Nickel et al., 2011), DisMult (Yang et al., 2014) and ConvKB (Nguyen et al., 2018). We evaluate the models in the “bern” and “filt” settings. The optimal configurations are illustrated in Appendix A Table 6.

The experimental results on FB15K-237 and WN18RR are given in Table 3. In each column, the top-1 result with bold marker and top-2-4 results with underline markers are given. Our presented models with $L_{AS}$ loss outperform the corresponding former models with $L_R$, $L_{RS}$ and $L_{SS}$ on all the metrics. The results also prove the effectiveness of our $L_{AS}$ loss. Detailed improved results for MRR (Compare in the best results) metric are as follows. On WN18RR, the results are increased by TransE 1.5%, TransH 1.1%, ComplEx 3.0% and RotatE 1.2% than corresponding $L_{SS}$ loss models. On FB15K237, the results are increased by TransE 0.8%, TransH-SS 1.2%, ComplEx-SS 2.9% and RotatE 0.6%.

| Models             | MR (%) | MRR (%) | Hits@1 (%) | Hits@3 (%) | Hits@10 (%) |
|--------------------|--------|---------|------------|------------|-------------|
| RESCAL             | 100.77 | 24.7    | 19.9       | 27.7       | 35.2        |
| DistMult           | 51.10  | 43.0    | 39.4       | 44.4       | 49.4        |
| ConvKB             | 1295   | 26.5    | 5.8        | 44.5       | 55.8        |
| TransE             | 3530   | 20.7    | 2.2        | 36.1       | 47.8        |
| TransE-RS          | 3415   | 20.8    | 2.3        | 36.3       | 47.8        |
| TransE-SS          | 3199   | 20.9    | 2.5        | 37.1       | 47.9        |
| TransE-CAS(ours)   | 1868   | 22.4    | 7.1        | 33.6       | 48.7        |
| TransE-IAS(ours)   | 3276   | 21.0    | 2.2        | 38.1       | 49.5        |
| TransH             | 3972   | 19.8    | 0.7        | 36.3       | 46.3        |
| TransH-SS          | 3422   | 20.0    | 1.0        | 37.3       | 47.8        |
| TransH-CAS(ours)   | 2890   | 21.2    | 2.4        | 37.9       | 47.8        |
| TransH-IAS(ours)   | 3145   | 21.1    | 0.8        | 38.7       | 49.6        |
| ComplEx            | 5246   | 40.1    | 36.2       | 42.5       | 47.1        |
| ComplEx-SS         | 5152   | 41.3    | 37.8       | 44.5       | 50.6        |
| ComplEx-CAS(ours)  | 4788   | 43.6    | 39.2       | 46.0       | 50.5        |
| ComplEx-IAS(ours)  | 4814   | 44.3    | 40.9       | 46.0       | 50.6        |
| RotatE            | 3735   | 47.1    | 42.3       | 48.7       | 56.4        |
| RotatE-CAS(ours)   | 3651   | 47.9    | 43.5       | 49.6       | 56.4        |
| RotatE-IAS(ours)   | 3862   | 48.3    | 46.7       | 50.2       | 57.0        |

Table 3: Evaluation results on WN18RR, FB15k-237 datasets. § donates trained with larger epoch, embedding dim and self-adversarial negative sampling (Sun et al., 2019).

TransH-SS 1.6% and ComplEx-SS 0.7%.

### 4.2 Triplet Classification

Triplet classification is a binary classification problem used to decide whether a given triplet $(h, r, t)$ is correct or not. This task is usually tested by trans-
lation models, but it is rarely validated by nonlinear models (Bordes et al., 2013; Dettmers et al., 2018). Therefore, in this experiment, we only test the series of the compared translation models. We use three datasets, WN11, FB13 and FB15K (see Table 1) 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 \(f_s(h, t)\) 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 loss framework \(L_{AS}\) used in TransE and TransH with baseline methods reported in Wang et al. (2014); Ji et al. (2015); Lin et al. (2015) who used the same datasets. TransE-SS and TransH-SS (Zhou et al., 2021) are retrained with the best configure in our framework. In the test phase, we need negative triplets for the binary classification evaluation. The datasets WN11 and FB13 released by NTN (Socher et al., 2013) with negative triplets. For FB15k, we construct the negative triplets following (Socher et al., 2013). The optimal configurations are illustrated in Appendix A Table 7.

The experimental results on triplet classification are shown in Table 4. In each column, the top-1 result with bold marker and top-2-3 results with underline markers are given. On WN11, models with \(L_{AS}\) all can reach an accuracy of 84%. On FB13, models with \(L_{AS}\) are comparable to former loss models. On FB15K, models with \(L_{AS}\) have significant improvement compared to former models, and TransH-CAS performs best resulting 91.6% accuracy among the compared models.

### 4.3 Discussion

**Impact of the hyper-parameters.** We analyze the impact of two hyper-parameters \(\mu_p\) (the upper score margin for all positive triplets) and \(\mu_n\) (the lower score margin for all negative triplets). On the WN18 dataset, we first select a fixed value of \(\mu_p\), and test the impact of different values of \(\mu_n = \mu_p + \{0.1, 0.25, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}\) on the experimental results. Figure 3 shows that good results can be obtained when \(\mu_p - \mu_n\) is in the range of 2-7. Compared with \(L_{SS}\), \(L_{AS}\) is more robust when \(\mu_p - \mu_n\) takes a larger value.

**Analysis of the convergence.** We analyze the convergence of \(L_{AS}\) and \(L_R, L_{RS}, L_{SS}\) with TransE model on the FB15K dataset. Figure 4a shows the convergence curve of different loss functions after normalization. From the figure, we can see that \(L_{AS}\) can converge more quickly and reach lower states. This phenomenon confirms that \(L_{AS}\) has a more definite convergence target, which promotes separability for positive and negative triplets.

**Analysis of the dynamic weight.** We analyze the mean valid weights of positive and negative triplets \((S_p - v_p > 0\) and \(S_p - \mu_p > 0\) for \(\alpha_p\), \(v_n - S_n > 0\) and \(\mu_p S_p > 0\) for \(\alpha_p\)). Figure 4b shows the dynamic changes of \(\alpha_p, \alpha_n\) of TransH on the WN18 dataset (\(i\) donates IAS, \(c\) donates CAS). Normally, the positive triplets are further away from optimization at the beginning, so the value of \(\alpha_p\) is larger. From Figure 4b we can see that the weight change of \(L_{IAS}\) is more sensitive than \(L_{CAS}\), and the overall weight dynamic changes of the two are closer. For practical applications, we recommend using the simpler \(L_{CAS}\) first, and \(L_{IAS}\) may bring some better results.

![Figure 3: The impact of hyper-parameter \(\mu_n - \mu_p\).](image)

![Figure 4: (a) Convergence of Loss Function. (b) Changes of dynamic weight](image)

### 5 Conclusion

In this paper, we propose a novel adaptive limit scoring loss framework for learning knowledge
graph embeddings. The key idea of our proposal adaptive scoring loss is to re-weight each triplet and highlight the less-optimized triplet scores. For the setting of dynamic weights, we propose constant adaptive and independent adaptive methods according to the positioning of the circle center. We apply our loss framework on several knowledge graph embedding models such as TransE, TransH, ComplEx and RotatE, and conduct experiments on WordNet and Freebase datasets with link prediction and triplet classification tasks. The experimental results show the superiority of our proposed method.

Acknowledgement

This work was supported in part by State Key Development Program Grand No. 2020YFB1708002, and NNSFC Grant No. 61971008.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In Proc. of ACM International Conference on Management of Data, pages 1247–1250.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In Proc. of Conference on Empirical Methods in Natural Language Processing, pages 615–620.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In Proc. of Artificial Intelligence and Statistics, pages 127–135.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In Proc. of Annual Conference on Neural Information Processing Systems, pages 1–9.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In Proc. of AAAI Conference on Artificial Intelligence, volume 32, pages 1811–1818.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proc. of ACM International Conference on Knowledge Discovery and Data Mining, pages 601–610.

Lucas Drumond, Steffen Rendle, and Lars Schmidt-Thieme. 2012. Predicting rdf triples in incomplete knowledge bases with tensor factorization. In Proc. of Annual ACM Symposium on Applied Computing, pages 326–331.

SM Shamimul Hasan, Donna Rivera, Xiao-Cheng Wu, Eric B Durbin, J Blair Christian, and Georgia Tourassi. 2020. Knowledge graph-enabled cancer data analytics. IEEE journal of biomedical and health informatics, 24(7):1952–1967.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In Proc. of ACM International Conference on Web Search and Data Mining, pages 105–113.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In Proc. of Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing, pages 687–696.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In Proc. of AAAI Conference on Artificial Intelligence, pages 985–991.

Xiaotian Jiang, Quan Wang, and Bin Wang. 2019. Adaptive convolution for multi-relational learning. In Proc. of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 978–987.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In Proc. of International Conference on Machine Learning, pages 2863–2872.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Klee, Sören Auer, et al. 2015. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web, 6(2):167–195.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In Proc. of AAAI Conference on Artificial Intelligence, volume 29, pages 2181–2187.

Shuqi Lu, Zhicheng Dou, Chenyuan Xiong, Xiaojie Wang, and Ji-Rong Wen. 2020. Knowledge enhanced personalized search. In Proc. of ACM International Conference on Research and Development in Information Retrieval, pages 709–718.

George A Miller. 1995. Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41.
Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. In Proc. of Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 460–466.

Tu Nguyen, Dinh Phung, et al. 2020. A relational memory-based embedding model for triple classification and search personalization. In Proc. of Annual Meeting of the Association for Computational Linguistics, pages 3429–3435.

Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In Proc. of North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 327–333.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In Proc. of AAAI Conference on Artificial Intelligence, volume 30, pages 1955–1961.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In Proc. of International Conference on Machine Learning, volume 11, pages 809–816.

Baoxu Shi and Tim Weninger. 2017. Proje: Embedding projection for knowledge graph completion. In Proc. of AAAI Conference on Artificial Intelligence, volume 31, pages 1236–1242.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In Proc. of Annual Conference on Neural Information Processing Systems, pages 926–934.

George Stoica, Otilia Stretcu, Emmanouil Antonios Platanios, Tom Mitchell, and Barnabás Póczos. 2020. Contextual parameter generation for knowledge graph link prediction. In Proc. of AAAI Conference on Artificial Intelligence, volume 34, pages 3000–3008.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In Proc. of International Conference on World Wide Web, pages 697–706.

Yifan Sun, Changmao Cheng, Yuhai Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. 2020. Circle loss: A unified perspective of pair similarity optimization. In Proc. of Computer Vision and Pattern Recognition, pages 6398–6407.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In Proc. of International Conference on Learning Representations, pages 1–18.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In Proc. of Workshop on Continuous Vector Space Models and their Compositionality, pages 57–66.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In Proc. of International Conference on Machine Learning, volume 48, pages 2071–2080.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In Proc. of AAAI Conference on Artificial Intelligence, volume 28, pages 1112–1119.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Learning multi-relational semantics using neural-embedding models. arXiv preprint arXiv:1411.4072.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In Proc. of International Conference on Learning Representations, pages 1–12.

Jinfa Yang, Yongjie Shi, Xin Tong, Robin Wang, Taiyan Chen, and Xianghua Ying. 2021. Improving knowledge graph embedding using affine transformations of entities corresponding to each relation. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 508–517.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In Proc. of ACM International Conference on Knowledge Discovery and Data Mining, pages 1006–1014.

Xiaofei Zhou, Lingfeng Niu, Qiannan Zhu, Xingquan Zhu, Ping Liu, Jianlong Tan, and Li Guo. 2021. Knowledge graph embedding by double limit scoring loss. IEEE Transactions on Knowledge and Data Engineering.

Xiaofei Zhou, Qiannan Zhu, Ping Liu, and Li Guo. 2017. Learning knowledge embeddings by combining limit-based scoring loss. In Proc. of ACM on Conference on Information and Knowledge Management, pages 1009–1018.

A Parameter Settings

Table 5 shows the parameter settings of TransE, TransH, ComplEx with adaptive limit scoring loss for link prediction on WN18, FB15K datasets. Table 6 shows the parameter settings of TransE, TransH, ComplEx, RotatE with adaptive Limit Scoring Loss for link prediction on the WN18NN,
FB15K237 datasets, where $t$ represents the sampling temperature for self-adversarial negative sampling. Table 7 shows the parameter settings of TransE, TransH with adaptive Limit Scoring Loss for triplet classification on the WN18, FB13 and FB15K datasets.

Table 6: Parameter Configurations for WN18RR and WN18

| WN18  | B   | m   | $\alpha$ | $\mu_p$ | $\mu_n$ | C/t |
|-------|-----|-----|----------|---------|---------|-----|
| TransE-CAS | 1000 | 200 | 0.00001 | 4.0 | 9.0 | - |
| TransE-IAS | 1000 | 100 | 0.00005 | 4.0 | 8.0 | - |
| TransH-CAS | 500 | 80 | 0.00005 | 4.0 | 9.0 | 0.0005 |
| TransH-IAS | 50 | 80 | 0.00005 | 3.0 | 7.0 | 0.0005 |
| ComplEx-CAS | 1000 | 200 | 0.00005 | 0.3 | 0.7 | - |
| ComplEx-IAS | 500 | 200 | 0.00005 | 0.1 | 0.7 | - |

| FB15k | B   | m   | $\alpha$ | $\mu_p$ | $\mu_n$ | C/t |
|-------|-----|-----|----------|---------|---------|-----|
| TransE-CAS | 1000 | 200 | 0.0001 | 6.0 | 6.5 | - |
| TransE-IAS | 1000 | 200 | 0.00005 | 6.0 | 7.0 | - |
| TransH-CAS | 1000 | 200 | 0.00001 | 10.0 | 11.0 | 0.0625 |
| TransH-IAS | 500 | 200 | 0.0001 | 7.0 | 8.0 | 0.0625 |
| ComplEx-CAS | 1000 | 200 | 0.00005 | 0.6 | 0.7 | - |
| ComplEx-IAS | 1000 | 200 | 0.00005 | 0.6 | 0.8 | - |

Table 5: Parameter Configurations for WN18 and FB15K

| WN18RR | B   | m   | $\alpha$ | $\mu_p$ | $\mu_n$ | C/t |
|--------|-----|-----|----------|---------|---------|-----|
| TransE-CAS | 50 | 50 | 0.00005 | 2.0 | 12.0 | - |
| TransE-IAS | 500 | 150 | 0.00005 | 5.0 | 10.0 | - |
| TransH-CAS | 200 | 50 | 0.005 | 3.0 | 10.0 | 0.0005 |
| TransH-IAS | 200 | 150 | 0.00005 | 5.0 | 10.0 | 0.0005 |
| ComplEx-CAS | 1000 | 200 | 0.00001 | 0.1 | 0.3 | - |
| ComplEx-IAS | 100 | 200 | 0.00001 | 0.1 | 0.5 | - |
| RotatE-CAS | 500 | 500 | 0.00001 | 1.0 | 4.0 | $\approx$0.5 |
| RotatE-IAS | 500 | 500 | 0.00001 | 1.0 | 4.0 | $\approx$0.5 |

| FB15k-237 | B   | m   | $\alpha$ | $\mu_p$ | $\mu_n$ | C/t |
|-----------|-----|-----|----------|---------|---------|-----|
| TransE-CAS | 1000 | 200 | 0.00005 | 7.0 | 9.0 | - |
| TransE-IAS | 500 | 200 | 0.00001 | 7.0 | 9.0 | - |
| TransH-CAS | 1000 | 200 | 0.00001 | 6.0 | 8.0 | 0.0625 |
| TransH-IAS | 1000 | 200 | 0.00001 | 6.0 | 8.0 | 0.0625 |
| ComplEx-CAS | 2000 | 200 | 0.00005 | 0.6 | 0.65 | - |
| ComplEx-IAS | 2000 | 200 | 0.00005 | 0.6 | 0.7 | - |
| RotatE-CAS | 1000 | 1000 | 0.0001 | 3.0 | 5.0 | $\approx$0.5 |
| RotatE-IAS | 1000 | 1000 | 0.00001 | 3.0 | 4.0 | $\approx$1.0 |

B Training Process

Training process of knowledge graph embedding models with adaptive scoring loss $L_{AS}$ is given in Algorithm 1. Where $\mathcal{G}$ donates a knowledge graph composed of several triplets; $N_e, N_r$ donate the number of entities and relations respectively; $d, k$ represent the embedding dimensions of entities and relations, usually $d = k$; $\mathbf{mE} \in \mathbb{R}^{N_e \times d}$, $\mathbf{mR} \in \mathbb{R}^{N_r \times k}$ donate the embedding of entities and relations respectively.

Algorithm 1: Learning knowledge graph embedding models with $L_{AS}$

Input: Positive training triplets $\mathcal{G} = \{(h,r,t)|h \in \mathcal{E}, r \in \mathcal{R}\}$, $\mathcal{E}$ and $\mathcal{R}$ are respectively the set of entities and relations. Negative training triplets $\mathcal{G}' = \emptyset$.

Output: Entity and relation embedding $\mathbf{mE}$ and $\mathbf{mR}$

Stage1: Initialization of Knowledge Graphs.

1. Entity embedding $\mathbf{mE} \leftarrow$ initialization ($N_e, d$);
2. Entity embedding $\mathbf{mR} \leftarrow$ initialization ($N_r, k$);
3. if initialization($a,b$) produces a matrix with size by initialized randomly or the results of basic models such as TransE (Bordes et al., 2013);

Stage2: Construct Negative Triplets.

for each $(h,r,t)$ in positive sample set $\mathcal{G}$ do

4. $(h',r,t')$ = generate_negative($h,r,t$) using unif/bern strategy in (Wang et al., 2014) for generating negative samples;
5. $\mathcal{G}' = \mathcal{G}' \cup (h',r,t')$
end

Stage3: Learning Embeddings of Entities and Relations.

for $e \leftarrow 1$ to MaxEpoch do

for $i \leftarrow 1$ to MaxSample do

6. $\mathcal{S}p_i = \text{sample\_batch}\left(\mathcal{G}, \mathcal{G}', B\right)$ // sample a mini-batch of size $B$ at random from positive and negative training samples;
7. Update entity and relation embeddings w.r.t. the gradients of $\sum_{(h,r,t),(h',r,t') \in \mathcal{S}p_i} \alpha_p |S_p - \mu_p| + \alpha_n |\mu_n - S_n|$;
8. Handle additional constraints or regularization terms;
end