Duality-Induced Regularizer for Tensor Factorization Based Knowledge Graph Completion

Zhanqiu Zhang Jianyu Cai Jie Wang
University of Science and Technology of China
{zzq96,jycai}@mail.ustc.edu.cn,jiewangx@ustc.edu.cn

Abstract

Tensor factorization based models have shown great power in knowledge graph completion (KGC). However, their performance usually suffers from the overfitting problem seriously. This motivates various regularizers—such as the squared Frobenius norm and tensor nuclear norm regularizers—while the limited applicability significantly limits their practical usage. To address this challenge, we propose a novel regularizer—namely, Duality-induced Regularizer (DURA)—which is not only effective in improving the performance of existing models but widely applicable to various methods. The major novelty of DURA is based on the observation that, for an existing tensor factorization based KGC model (primal), there is often another distance based KGC model (dual) closely associated with it. Experiments show that DURA yields consistent and significant improvements on benchmarks.

1 Introduction

Knowledge graphs contain quantities of factual triplets, which represent structured human knowledge. In the past few years, knowledge graphs have made great achievements in many areas, such as natural language processing [37], question answering [13], recommendation systems [30], and computer vision [18]. Although commonly used knowledge graphs usually contain billions of triplets, they still suffer from the incompleteness problem that a lot of factual triplets are missing. Due to the large scale of knowledge graphs, it is impractical to find all valid triplets manually. Therefore, knowledge graph completion (KGC)—which aims to predict missing links between entities based on known links automatically—has attracted much attention recently.

Distance based (DB) models and tensor factorization based (TFB) models are two important categories of KGC models. DB models use the Minkowski distance to measure the plausibility of a triplet. Although they can achieve state-of-the-art performance, many of them still have difficulty in modeling complex relation patterns, such as one-to-many and many-to-one relations [16, 33]. TFB models treat knowledge graphs as partially observed third-order binary tensors and formulate KGC as a tensor completion problem. Theoretically, these models are highly expressive and can well handle complex relations. However, their performance usually suffers from the overfitting problem seriously and consequently cannot achieve state-of-the-art.

To tackle the overfitting problem of TFB models, researchers propose various regularizers. The squared Frobenius norm regularizer is a popular one that applies to various models [22, 34, 28]. However, experiments show that it may decrease performance for some models (e.g., RESCAL) [23]. More recently, motivated by the great success of the matrix trace norm in the matrix completion problem [25, 5], Lacroix et al. [15] propose a tensor nuclear $p$-norm regularizer. It gains significant improvements against the squared Frobenius norm regularizer. However, it is only suitable for canonical polyadic (CP) decomposition [12] based models, such as CP and ComplEx [28], but

*Corresponding author.

not appropriate for a more general class of models, such as RESCAL \cite{bordes:2013}. Therefore, it is still challenging to find a regularizer that is both widely applicable and effective.

In this paper, we propose a novel regularizer for tensor factorization based KGC models—namely, DUality-induced ReguARizer (DURA). The major novelty of DURA is based on the observation called duality—for an existing tensor factorization based KGC model (primal), there is often another distance based KGC model closely associated with it (dual). The duality can be derived by expanding the squared score functions of the associated distance based models. Then, the cross-term in the expansion is exactly a tensor factorization based KGC model, and the squared terms in it give us a regularizer. Using DURA, we can preserve the expressiveness of tensor factorization based KGC models and prevent them from the overfitting problem. DURA is widely applicable to various tensor factorization based models, including CP, ComplEx, and RESCAL. Experiments show that, DURA yields consistent and significant improvements on datasets for the knowledge graph completion task. It is worth noting that, when incorporated with DURA, RESCAL \cite{bordes:2013}—which is one of the first knowledge graph completion models—performs comparably to state-of-the-art methods and even beats them on several benchmarks.

2 Preliminaries

In this section, we review the background of this paper in Section 2.1 and introduce the notations used throughout this paper in Section 2.2.

2.1 Background

Knowledge Graph Given a set $E$ of entities and a set $R$ of relations, a knowledge graph $K = \{(e_i, r_j, e_k)\} \subset E \times R \times E$ is a set of triplets, where $e_i$ and $r_j$ are the $i$-th entity and $j$-th relation, respectively. Usually, $e_i$ and $e_k$ are also called the head entity and the tail entity, respectively.

Knowledge Graph Completion (KGC) The goal of KGC is to predict valid but unobserved triplets based on the known triplets in $K$. KGC models contain two important categories: distance based models and tensor factorization based models, both of which are knowledge graph embedding (KGE) methods. KGE models associate each entity $e_i \in E$ and relation $r_j \in R$ with an embedding (may be real or complex vectors, matrices, and tensors) $e_i$ and $r_j$. Generally, they define a score function $s : E \times R \times E \rightarrow \mathbb{R}$ to associate a score $s(e_i, r_j, e_k)$ with each potential triplet $(e_i, r_j, e_k) \in E \times R \times E$. The scores measure the plausibility of triplets. For a query $(e_i, r_j, ?)$, KGE models first fill the blank with each entity in the knowledge graphs and then score the resulted triplets. Valid triplets are expected to have higher scores than invalid triplets.

Distance Based (DB) KGC Models DB models define the score function $s$ with the Minkowski distance. That is, the score functions have the formulation of $s(e_i, r_j, e_k) = -\|\Gamma(e_i, r_j, e_k)\|_p$, where $\Gamma$ is a model-specific function. Equivalently, we can also use a squared score function $s(e_i, r_j, e_k) = -\|\Gamma(e_i, r_j, e_k)\|^2_p$.

Tensor Factorization Based (TFB) KGC Models TFB models regard a knowledge graph as a third-order binary tensor $\mathcal{X} \in \{0, 1\}^{E \times |R| \times |\mathcal{E}|}$. The $(i, j, k)$ entry $\mathcal{X}_{ijk} = 1$ if $(e_i, r_j, e_k)$ is valid otherwise $\mathcal{X}_{ijk} = 0$. Suppose that $\mathcal{X}_j$ denotes the $j$-th frontal slice of $\mathcal{X}$, i.e., the adjacency matrix of the $j$-th relation. Usually, a TFB KGC model factorizes $\mathcal{X}_j$ as $\mathcal{X}_j \approx \text{Re}(\mathcal{H}_j \mathbf{T}_j^\top)$, where the $i$-th ($k$-th) row of $\mathcal{H}$ ($\mathbf{T}$) is $e_i$ ($e_k$), $\mathcal{R}_j$ is a matrix representing relation $r_j$, $\text{Re} (\cdot)$ and $\mathcal{R}^\top$ are the real part and the conjugate of a complex matrix, respectively. That is, the score functions are defined as $s(e_i, r_j, e_k) = \text{Re}(\mathcal{R}_j \hat{e}_i \hat{e}_k^\top)$. Note that the real part and the conjugate of a real matrix are itself. Then, the aim of TFB models is to seek matrices $\mathbf{H}_1, \ldots, \mathbf{H}_{|R|}$, $\mathbf{T}$, such that $\text{Re}(\mathcal{H}_j \mathbf{T}_j^\top)$ can approximate $\mathcal{X}_j$. Let $\hat{\mathcal{X}}_j = \text{Re}(\mathcal{H} \mathcal{R}^\top)$ and $\hat{\mathcal{X}}$ be a tensor of which the $j$-th frontal slice is $\hat{\mathcal{X}}_j$. The regularized formulation of a tensor factorization based model can be written as

$$
\min_{\hat{\mathcal{X}}_1, \ldots, \hat{\mathcal{X}}_{|R|}} \sum_{j=1}^{|R|} L(\mathcal{X}_j, \hat{\mathcal{X}}_j) + \lambda g(\hat{\mathcal{X}}),
$$

(1)

where $\lambda > 0$ is a fixed parameter, $L(\mathcal{X}_j, \hat{\mathcal{X}}_j)$ measures the discrepancy between $\mathcal{X}_j$ and $\hat{\mathcal{X}}_j$, and $g$ is the regularization function.
We use $h_i \in \mathcal{E}$ and $t_k \in \mathcal{E}$ to distinguish head and tail entities. Let $\| \cdot \|_1$, $\| \cdot \|_2$, and $\| \cdot \|_F$ denote the $L_1$ norm, the $L_2$ norm, and the Frobenius norm of matrices or vectors. We use $\langle \cdot, \cdot \rangle$ to represent the inner products of two real or complex vectors. Specifically, if $u, v \in \mathbb{C}^{1 \times n}$ are two row vectors in the complex space, then the inner product is defined as $\langle u, v \rangle = uv^\top$.

## 3 Related Work

Knowledge graph completion (KGC) models include rule-based methods \cite{9,35}, KGE methods, and hybrid methods \cite{10}. This work is related to KGE methods \cite{4,28,21,36}. More specifically, it is related to distance based KGE models and tensor factorization based KGE models.

Distance based models describe relations as relational maps between head and tail entities. Then, they use the Minkowski distance to measure the plausibility of a given triplet. For example, TransE \cite{4} and its variants \cite{33,16} represent relations as translations in vector spaces. They assume that a valid triplet $(h_i, r_j, t_k)$ satisfies $h_i + r_j \approx t_k$, where $h_i$, $r_j$, and $t_k$ denote the entity embeddings corresponding to the head, relation, and tail entities, respectively. Structured embedding (SE) \cite{1} uses linear maps to represent relations. Its score function is defined as $s(h_i, r_j, t_k) = -\|R^j_i h_i - R^j_i t_k\|_1$, where $h_i, r_j, t_k \in \mathbb{C}^k$ and $\|r\|_1 = 1$. ModE \cite{38} assumes that $R^j_i$ is diagonal and $R^j_i$ is an identity matrix. It shares a similar score function $s(h_i, r_j, t_k) = -\|h_i \circ r_j - t_k\|_1$ with RotatE but $h_i, r_j, t_k \in \mathbb{R}^k$.

Tensor factorization based models formulate the KGC task as a third-order binary tensor completion problem. RESCAL \cite{22} factorizes the $j$-th frontal slice of $X_j \approx ARAX^\top$, in which embeddings of head and tail entities are from the same space. As the relation specific matrices contain lots of parameters, RESCAL is prone to overfitting. DistMult \cite{21} simplifies the matrix $R_j$ in RESCAL to be diagonal, while it sacrifices the expressiveness of models and can only handle symmetric relations. In order to model asymmetric relations, ComplEx \cite{28} extends DistMult to complex embeddings. Both DistMult and ComplEx can be regarded as variants of CP decomposition \cite{12}, which are in real and complex vector spaces, respectively.

Tensor factorization based (TFB) KGC models usually suffer from overfitting problem seriously, which motivates various regularizers. In the original papers of TFB models, the authors usually use the squared Frobenius norm ($L_2$ norm) regularizer \cite{22,34,28}. This regularizer cannot bring satisfying improvements. Consequently, TFB models do not gain comparable performance to distance based models \cite{26,38}. More recently, Lacroix et al. \cite{15} propose to use the tensor nuclear 3-norm \cite{8} (N3) as a regularizer, which brings more significant improvements than the squared Frobenius norm regularizer. However, it is designed for the CP-like models, such as CP and ComplEx, and not suitable for more general models such as RESCAL. Moreover, some regularization methods aim to leverage external background knowledge \cite{19,7,20}. For example, to model equivalence and inversion axioms, Minervini et al. \cite{19} impose a set of model-dependent soft constraints on the predicate embeddings. Ding et al. \cite{7} use non-negativity constraints on entity embeddings and approximate entailment constraints on relation embeddings to impose prior beliefs upon the structure of the embeddings space.

## 4 Methods

In this section, we introduce a novel regularizer—DUality-induced RegulArizer (DURA)—for tensor factorization based knowledge graph completion. We first introduce basic DURA in Section 4.1 and explain why it is effective in Section 4.2. Then, we introduce DURA in Section 4.3. Finally, we give a theoretical analysis for DURA under some special cases in Section 4.4.

### 4.1 Basic DURA

Consider the knowledge graph completion problem $(h_i, r_j, ?)$. That is, we are given the head entity and the relation, aiming to predict the tail entity. Suppose that $f_j(i, k)$ measures the plausibility of a
Figure 1: An illustration of why basic DURA can improve the performance of TFB models when the embedding dimensions are 2. Suppose that triplets \((h_i, r_j, t_k)\) \((k = 1, 2, \ldots, n)\) are valid. (a) Figure 1a demonstrates that tail entities connected to a head entity through the same relation should have similar embeddings. Suppose that we know a head entity \(h_i\) and a relation \(r_j\), and our aim is to predict the tail entity. If \(r_j\) is a one-to-many relation, i.e., there exist two entities \(t_1\) and \(t_2\) such that \(h_i, r_j, t_1\) and \(h_i, r_j, t_2\) are valid. (b) Figure 1b shows that TFB models without regularization can get the same score even though the embeddings of \(t_k\) are dissimilar. (c) Figure 1c shows that with DURA, embeddings of \(t_k\) are encouraged to locate in a small region.

Given triplet \((h_i, r_j, t_k)\), i.e., \(f_j(i, k) = s(h_i, r_j, t_k)\). Then the score function of a TFB model is

\[
f_j(i, k) = \text{Re}(h_i, R_j, t_k) = \text{Re}(\langle h_i, \overline{R}_j, t_k \rangle).
\]

(2)

It first maps the entity embeddings \(h_i\) by a linear transformation \(\overline{R}_j\), and then uses the real part of an inner product to measure the similarity between \(h_i, \overline{R}_j\) and \(t_k\). Notice that another commonly used similarity measure—the squared Euclidean distance—can replace the inner product similarity in Equation (2). We can obtain an associated distance based model formulated as

\[
f_j^E(i, k) = −∥h_i, \overline{R}_j − t_k∥_2^2.
\]

(3)

Therefore, there exists a duality: for an existing tensor factorization based KGC model \((\text{primal})\), there is often another distance based KGC model \((\text{dual})\) closely associated with it.

Specifically, the relationship between the primal and the dual can be formulated as

\[
f_j^E(i, k) = −∥h_i, \overline{R}_j − t_k∥_2^2
\]

\[
= −∥h_i, \overline{R}_j∥_2^2 − ∥t_k∥_2^2 + 2\text{Re}(\langle h_i, \overline{R}_j, t_k \rangle)
\]

(4)

\[
= 2f_j(i, k) − ∥h_i, \overline{R}_j∥_2^2 − ∥t_k∥_2^2.
\]

Usually, we expect \(f_j^E(i, k)\) and \(f_j(i, k)\) to be higher for all valid triplets \((h_i, r_j, t_k)\) than those for invalid triplets. Suppose that \(S\) is the set that contains all valid triplets. Then, for triplets in \(S\), we have that

\[
\max f_j^E(i, k) = \min −f_j^E(i, k)
\]

\[
= \min −2f_j(i, k) + ∥h_i, \overline{R}_j∥_2^2 + ∥t_k∥_2^2.
\]

(5)

By noticing that \(\min −2f_j(i, k) = \max 2f_j(i, k)\) is exactly the aim of a TFB model, the duality induces a regularizer for tensor factorization based KGC models, i.e.,

\[
\sum_{(h_i, r_j, t_k) ∈ S} ∥h_i, \overline{R}_j∥_2^2 + ∥t_k∥_2^2,
\]

(6)

which is called basic DURA.

4.2 Why Basic DURA Helps

In this section, we demonstrate that basic DURA encourages tail entities connected to a head entity through the same relation to have similar embeddings, which accounts for its effectiveness in improving performance of TFB models.

First, we claim that tail entities connected to a head entity through the same relation should have similar embeddings. Suppose that we know a head entity \(h_i\) and a relation \(r_j\), and our aim is to predict the tail entity. If \(r_j\) is a one-to-many relation, i.e., there exist two entities \(t_1\) and \(t_2\) such...
that both \((h_i, r_j, t_1)\) and \((h_i, r_j, t_2)\) are valid, then we expect that \(t_1\) and \(t_2\) have similar semantics. For example, if two triplets \((\text{fleid, include, tigers})\) and \((\text{fleid, include, lions})\) are valid, then \text{tigers} and \text{lions} should have similar semantics. Further, we expect that entities with similar semantics have similar embeddings. In this way, if we have known that \((\text{tigers, is, mammals})\) is valid, then we can predict that \((\text{lions, is, mammals})\) is also valid. See Figure 1a for an illustration of the prediction process.

However, TFB models fail to achieve the above goal. As shown in Figure 1b, suppose that we have known \(h, \bar{R}_t\) when the embedding dimension is 2. Then, we can get the same score \(s(h_i, r_j, t_k)\) for \(k = 1, 2, \ldots, n\) so long as \(t_k\) lies on the same line perpendicular to \(h, \bar{R}_j\). Generally, the entities \(t_1\) and \(t_2\) have similar semantics. However, their embeddings \(t_1\) and \(t_2\) can even be orthogonal, which means that the two embeddings are dissimilar. Therefore, the performance of TFB models for knowledge graph completion is usually unsatisfying.

By Equation (5), we know that basic DURA constrains the distance between \(h, \bar{R}_j\) and \(t_k\). When \(h\) and \(\bar{R}_j\) are known, \(t_k\) lies in a small region (see Figure 1c) and we verify this claim in Section 5.4. Therefore, tail entities connected to a head entity through the same relation will have similar embeddings, which is beneficial to the prediction of unknown triplets.

### 4.3 DURA

Basic DURA encourages tail entities with similar semantics to have similar embeddings. However, it cannot handle the case that head entities have similar semantics.

Suppose that two triplets \((\text{tigers, is, mammals})\) and \((\text{lions, is, mammals})\) are valid. Similar to the discussion in Section 4.2, we expect that \text{tigers} and \text{lions} have similar semantics and thus have similar embeddings. If we further know that \((\text{fleid, include, tigers})\) is valid, we can predict that \((\text{fleid, include, lions})\) is valid. However, basic DURA cannot handle the case. Let \(h_1, h_2, t_1, t_2, \bar{R}_1\) and \(\bar{R}_2\) be the embeddings of \text{tigers, lions, mammals}, and is, respectively. Then, \(Re(\bar{h}_1 \bar{R}_1^t t_1^t)\) and \(Re(\bar{h}_2 \bar{R}_2^t t_2^t)\) can be equal even if \(h_1\) and \(h_2\) are orthogonal, as long as \(h_1 \bar{R}_1 = h_2 \bar{R}_1\).

To tackle the above issue, noticing that \(Re(\bar{h}_1 \bar{R}_1^t t_1^t) = Re(\bar{i}_k \bar{R}_1^t h_1^t)\), we define another dual distance based KGC model \[ f^E_j(i, k) = -\|t_k \bar{R}_j^T - h_i\|^2. \]

Then, similar to the derivation in Equation (5), the duality induces a regularizer given by \[ \sum_{\langle h, r, t_k \rangle \in S} \|t_k \bar{R}_j^T\|^2 + \|h_i\|^2. \] (7)

When a TFB model is incorporated with regularizer (7), head entities with similar semantics will have similar embeddings.

Finally, combining the regularizer (5) and (7), DURA has the form of \[ \sum_{\langle h, r, t_k \rangle \in S} \left[\|h_i \bar{R}_j\|^2 + \|t_k\|^2 + \|t_k \bar{R}_j^T\|^2 + \|h_i\|^2\right]. \] (8)

### 4.4 Theoretic Analysis for Diagonal Relation Matrices

If we further relax the summation condition in the regularizer (8) to all possible entities and relations, we can write DURA as:

\[ |\mathcal{E}| \sum_{j = 1}^{\mathcal{R}} \left( \|H \bar{R}_j\|^2 + \|T\|^2 + \|TR_j^T\|^2 + \|H\|^2 \right), \] (9)

where \(|\mathcal{E}|\) and \(|\mathcal{R}|\) are the number of entities and relations, respectively.

In the rest of this section, we use the same definitions of \(\hat{X}_j\) and \(\hat{X}\) as in the problem (1). When the relation embedding matrices \(R\) are diagonal in \(\mathbb{R}\) or \(\mathbb{C}\) as in CP or ComplEx, the formulation (9) gives an upper bound to the tensor nuclear 2-norm of \(\hat{X}\), which is an extension of trace norm regularizers in matrix completion. To simplify the notations, we take CP as an example, in which all involved embeddings are real. The conclusion in complex space can be analogized accordingly.
Definition 1 (Friedland & Lim [8]). The nuclear 2-norm of a 3D tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is
\[
\|\mathcal{A}\|_* = \min \left\{ \sum_{i=1}^{r} \|u_{1,i}\|_2 \|u_{2,i}\|_2 \|u_{3,i}\|_2 : \mathcal{A} = \sum_{i=1}^{r} u_{1,i} \otimes u_{2,i} \otimes u_{3,i}, r \in \mathbb{N} \right\},
\]
where $u_{k,i} \in \mathbb{R}^{n_k}$ for $k = 1, ..., 3$, $i = 1, ..., r$, and $\otimes$ denotes the outer product.

For notation convenience, we define a relation matrix $\tilde{R} \in \mathbb{R}^{\vert \mathcal{R} \vert \times D}$, of which the $j$-th row consists of the diagonal entries of $R_j$. That is, $\tilde{R}(j, d) = R_j(d, d)$, where $R(j, i)$ represents the entry in the $i$-th row and $j$-th column of the matrix $R$.

In the knowledge graph completion problem, the tensor nuclear 2-norm of $\hat{X}$ is
\[
\|\hat{X}\|_* = \min \left\{ \sum_{d=1}^{D} \|h_d\|_2 \|r_d\|_2 \|t_d\|_2 : \hat{X} = \sum_{d=1}^{D} h_d \otimes r_d \otimes t_d \right\},
\]
where $D$ is the embedding dimension, $h_d$, $r_d$, and $t_d$ are the $d$-th columns of $H$, $\tilde{R}$, and $T$.

For DURA in (9), we have the following theorem.

**Theorem 1.** Suppose that $X_j = HR_jT_j^T$ for $j = 1, 2, \ldots, \vert \mathcal{R} \vert$, where $H, T, R_j$ are real matrices and $R_j$ is diagonal. Then, the following equation holds
\[
\min_{X_j=HR_jT_j^T} \frac{1}{\sqrt{\vert \mathcal{R} \vert}} \sum_{j=1}^{\vert \mathcal{R} \vert} \left( \|HR_j\|_F^2 + \|T\|_F^2 + \|TR_j^T\|_F^2 + \|H\|_F^2 \right) = \|\hat{X}\|_*.
\]

The minimization attains if and only if $\|h_d\|_2 \|r_d\|_2 = \frac{\sqrt{\vert \mathcal{R} \vert}}{\|t_d\|_2}$ and $\|t_d\|_2 \|r_d\|_2 = \sqrt{\vert \mathcal{R} \vert} \|h_d\|_2$, $\forall d \in \{1, 2, \ldots, D\}$, where $h_d$, $r_d$, and $t_d$ are the $d$-th columns of $H$, $\tilde{R}$, and $T$, respectively.

**Proof.** See the supplementary material.

Therefore, DURA in (9) gives an upper bound to the tensor nuclear 2-norm, which is a tensor analog to the matrix trace norm.

**Remark** DURA in (8) is actually a weighted version of the one in (9), in which the regularization terms corresponding to the sampled valid triplets. As shown in Srebro & Salakhutdinov [24] and Lacroix et al. [15], the weighted versions of regularizers usually outperform the unweighted regularizer when entries of the matrix or tensor are sampled non-uniformly. Therefore, in the experiments, we implement DURA in a weighted way as in (8).

## 5 Experiments

In this section, we introduce the experimental settings in Section 5.1 and show the effectiveness of DURA in Section 5.2. We compare DURA to other regularizers in Section 5.3 and visualize the entity embeddings in Section 5.4. Finally, we analyze the sparsity induced by DURA in Section 5.5. The code of HAKE is available on GitHub at https://github.com/MIRALab-USTC/KGE-DURA.

### 5.1 Experimental Settings

We consider three public knowledge graph datasets—WN18RR [27], FB15k-237 [6], and YAGO3-10 [17] for the knowledge graph completion task, which have been divided into training, validation, and testing set in previous works. The statistics of these datasets are shown in Table 1. WN18RR, FB15k-237, and YAGO3-10 are extracted from WN18 [14], FB15k [4], and YAGO3 [17], respectively. Toutanova & Chen [27] and Dettmers et al. [6] indicated the test set leakage problem in WN18 and FB15k, where some
test triplets may appear in the training dataset in the form of reciprocal relations. They created WN18RR and FB15k-237 to avoid the test set leakage problem, and we use them as the benchmark datasets. We use MRR and Hits@N (H@N) as evaluation metrics. For more details of training and evaluation protocols, please refer to the supplementary material.

Moreover, we find it better to assign different weights for the parts involved with relations. That is, the optimization problem has the form of

$$
\min \sum_{(e_i, r_j, e_k) \in \mathcal{S}} \left[ \ell_{ijk}(\mathbf{H}, \mathbf{R}_1, \ldots, \mathbf{R}_J, \mathbf{T}) \right] + \lambda (\lambda_1 (\|\mathbf{h}_i\|_2^2 + \|\mathbf{t}_k\|_2^2) + \lambda_2 (\|\mathbf{h}_i \mathbf{R}_j\|_2^2 + \|\mathbf{t}_k \mathbf{R}_j^\top\|_2^2))
$$

where $\lambda, \lambda_1, \lambda_2 > 0$ are fixed hyperparameters. We search $\lambda$ in $\{0.005, 0.01, 0.05, 0.1, 0.5\}$ and $\lambda_1, \lambda_2$ in $\{0.5, 1.0, 1.5, 2.0\}$.

Table 2: Evaluation results on WN18RR, FB15k-237 and YAGO3-10 datasets. We reimplement CP, DistMult, ComplEx, and RESCAL using the “reciprocal” setting [15, 14], which leads to better results than the reported results in the original paper.

<table>
<thead>
<tr>
<th></th>
<th>WN18RR</th>
<th>FB15k-237</th>
<th>YAGO3-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
<td>H@1</td>
<td>H@10</td>
</tr>
<tr>
<td>RotatE</td>
<td>.476</td>
<td>.428</td>
<td>.571</td>
</tr>
<tr>
<td>MuRP</td>
<td>.481</td>
<td>.440</td>
<td>.566</td>
</tr>
<tr>
<td>HAKE</td>
<td>.497</td>
<td>.452</td>
<td>.582</td>
</tr>
<tr>
<td>TuckER</td>
<td>.470</td>
<td>.443</td>
<td>.526</td>
</tr>
<tr>
<td>CP</td>
<td>.438</td>
<td>.414</td>
<td>.485</td>
</tr>
<tr>
<td>RESCAL</td>
<td>.455</td>
<td>.419</td>
<td>.493</td>
</tr>
<tr>
<td>ComplEx</td>
<td>.460</td>
<td>.428</td>
<td>.522</td>
</tr>
<tr>
<td>CP-DURA</td>
<td>.478</td>
<td>.441</td>
<td>.552</td>
</tr>
<tr>
<td>RESCAL-DURA</td>
<td><strong>.498</strong></td>
<td><strong>.455</strong></td>
<td><strong>.577</strong></td>
</tr>
<tr>
<td>ComplEx-DURA</td>
<td>.491</td>
<td>.449</td>
<td>.571</td>
</tr>
</tbody>
</table>

5.2 Main Results

In this section, we compare the performance of DURA against several state-of-the-art KGC models, including CP [12], RESCAL [22], ComplEx [28], TuckER [2] and some DB models: RotatE [26], MuRP [1], and HAKE [38].

Table 2 shows the effectiveness of DURA. RESCAL-DURA and ComplEx-DURA perform competitively with the SOTA DB models. RESCAL-DURA outperforms all the compared DB models in terms of MRR and H@1. Note that we reimplement CP, ComplEx, and RESCAL under the “reciprocal” setting [14, 15], and obtain better results than the reported performance in the original papers. Overall, TFB models with DURA significantly outperform those without DURA, which shows its effectiveness in preventing models from overfitting.

Generally, models with more parameters and datasets with smaller sizes imply a larger risk of overfitting. Among the three datasets, WN18RR has the smallest size of only 11 kinds of relations and around 80k training samples. Therefore, the improvements brought by DURA on WN18RR are expected to be larger compared with other datasets, which is consistent with the experiments. As stated in Wang et al. [31], RESCAL is a more expressive model, but it is prone to overfit on small- and medium-sized datasets because it represents relations with much more parameters. For example, on WN18RR dataset, RESCAL gets an H@10 score of 0.493, which is lower than ComplEx (0.522). The advantage of its expressiveness does not show up at all. However, incorporated with DURA, RESCAL gets an 8.4% improvement on H@10 and finally attains 0.577, which outperforms all compared models. On larger datasets such as YAGO3-10, overfitting also exists but may be non-significant. Nonetheless, DURA still leads to consistent improvement, demonstrating the ability of DURA to prevent models from overfitting.
We implement both the squared Frobenius norm (FRO) and N3 regularizers in the weighted way as widely applicable than N3 and more effective than the squared Frobenius norm regularizer.

Then, we use T-SNE to visualize the answers' embeddings generated by CP and CP-DURA. Figure 2 shows the visualization results. Each entity is represented by a 2D point and points in the same color represent tail entities with the same context. Figure 2(b) shows that, with DURA, entities with the same (hi, rj) contexts are indeed being assigned more similar representations, which verifies the claims in Section 4.2.

In this section, we compare DURA to the popular squared Frobenius norm regularizer and the recent tensor nuclear 3-norm (N3) regularizer [15]. The squared Frobenius norm regularizer is given by \( g(\hat{X}) = \|H \|^2_F + \|T \|^2_F + \sum_{j=1}^{R} \|R_j \|^2_F \). N3 regularizer is given by \( g(\hat{X}) = \sum_{d=1}^{D} (\|h_d\|^3 + \|r_d\|^3 + \|t_d\|^3) \), where \( \| \cdot \|_3 \) denotes L3 norm of vectors.

We implement both the squared Frobenius norm (FRO) and N3 regularizers in the weighted way as stated in Lacroix et al. [15]. Table 3 shows the performance of the three regularizers on three popular models: CP, ComplEx, and RESCAL. Note that when the TFB model is RESCAL, we only compare DURA to the squared Frobenius norm regularization as N3 does not apply to it.

For CP and ComplEx, DURA brings consistent improvements compared to FRO and N3 on all datasets. Specifically, on FB15k-237, compared to CP-N3, CP-DURA gets an improvement of 0.013 in terms of MRR. Even for the previous state-of-the-art TFB model ComplEx, DURA brings further improvements against the N3 regularizer. Incorporated with FRO, RESCAL performs worse than the vanilla model, which is consistent with the results in Ruffinelli et al. [23]. However, RESCAL-DURA brings significant improvements against RESCAL. All the results demonstrate that DURA is more widely applicable than N3 and more effective than the squared Frobenius norm regularizer.

5.3 Comparison to Other Regularizers

In this section, we visualize the tail entity embeddings using T-SNE [29] to show that DURA encourages tail entities with similar semantics to have similar embeddings.

Suppose that \((h_i, r_j)\) is a query, where \(h_i\) and \(r_j\) are head entities and relations, respectively. An entity \(t_k\) is an answer to a query \((h_i, r_j)\) if \((h_i, r_j, t_k)\) is valid. We randomly selected 10 queries in FB15k-237, each of which has more than 50 answers. Then, we use T-SNE to visualize the answers’ embeddings generated by CP and CP-DURA. Figure 2 shows the visualization results. Each entity is represented by a 2D point and points in the same color represent tail entities with the same \((h_i, r_j)\) context (i.e. query). Figure 2 shows that, with DURA, entities with the same \((h_i, r_j)\) contexts are indeed being assigned more similar representations, which verifies the claims in Section 4.2.

Table 3: Comparison between DURA, the squared Frobenius norm (FRO), and the nuclear 3-norm (N3) regularizers. Results of * are taken from Lacroix et al. [15]. CP-N3 and ComplEx-N3 are re-implemented and their performances are better than the reported results in Lacroix et al. [15]. The best performance on each model are marked in bold.

<table>
<thead>
<tr>
<th></th>
<th>WN18RR</th>
<th>FB15k-237</th>
<th>YAGO3-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
<td>H@1</td>
<td>H@10</td>
</tr>
<tr>
<td>CP-FRO*</td>
<td>.460</td>
<td>-</td>
<td>.480</td>
</tr>
<tr>
<td>CP-DURA</td>
<td>.478</td>
<td>.441</td>
<td>.552</td>
</tr>
<tr>
<td>ComplEx-FRO*</td>
<td>.470</td>
<td>-</td>
<td>.540</td>
</tr>
<tr>
<td>ComplEx-N3</td>
<td>.489</td>
<td>.443</td>
<td>.580</td>
</tr>
<tr>
<td>ComplEx-DURA</td>
<td>.491</td>
<td>.449</td>
<td>.571</td>
</tr>
<tr>
<td>RESCAL-FRO</td>
<td>.397</td>
<td>.363</td>
<td>.452</td>
</tr>
<tr>
<td>RESCAL-DURA</td>
<td>.498</td>
<td>.455</td>
<td>.577</td>
</tr>
</tbody>
</table>
5.5 Sparsity Analysis

As real-world knowledge graphs usually contain billions of entities, the storage of entity embeddings faces severe challenges. Intuitively, if embeddings are sparse, that is, most of the entries are zero, we can store them with less storage. Therefore, the sparsity of the generated entity embeddings becomes crucial for real-world applications. In this part, we analyze the sparsity of embeddings induced by different regularizers.

Generally, there are few entries of entity embeddings that are exactly equal to 0 after training, which means that it is hard to obtain sparse entity embeddings directly. However, when we score triplets using the trained model, the embedding entries with values close to 0 will have minor contributions to the score of a triplet. If we set the embedding entries close to 0 to be exactly 0, we can transform embeddings into sparse ones. Thus, there is a trade-off between sparsity and performance decrement.

We define the following $\lambda$-sparsity to indicate the proportion of entries that are close to zero:

$$s_\lambda = \frac{\sum_{i=1}^{I} \sum_{d=1}^{D} I\{|x|<\lambda\}(E_{id})}{I \times D},$$

where $E \in \mathbb{R}^{I \times D}$ is the entity embedding matrix, $E_{id}$ is the entry in the $i$-th row and $d$-th column of $E$, $I$ is the number of entities, $D$ is the embedding dimension, and $I_C(x)$ is the indicator function that takes value of 1 if $x \in C$ or otherwise the value of 0.

To generate sparse version entity embeddings, following Equation (10), we select all the entries of entity embeddings—of which the absolute value are less than a threshold $\lambda$—and set them to be 0. Note that for any given $s_\lambda$, we can always find a proper threshold $\lambda$ to approximate it, as the formula is increasing with respect to $\lambda$. Then, we evaluate the quality of sparse version entity embeddings on the knowledge graph completion task. Figure 3 shows the effect of entity embeddings’ $\lambda$-sparsity on MRR. Results in the figure show that DURA causes much gentler performance decrement as the embedding sparsity increases. In Figure 3a incorporated with DURA, CP maintains MRR of 0.366 unchanged even when 60% entries are set to 0. More surprisingly, when the sparsity reaches 70%, CP-DURA can still outperform CP-N3 with zero sparsity. For RESCAL, when setting 80% entries to be 0, RESCAL-DURA still has the MRR of 0.341, which significantly outperforms vanilla RESCAL, whose MRR has decreased from 0.352 to 0.286. In a word, incorporating with DURA regularizer, the performance of CP and RESCAL remains comparable to the state-of-the-art models, even when 70% of entity embeddings’ entries are set to 0.

Following Han et al. [11], we store the sparse version embedding matrices using compressed sparse row (CSR) format or compressed sparse column (CSC) format, which requires $2a + n + 1$ numbers, where $a$ is the number of non-zero elements and $n$ is the number of rows or columns. Experiments show that DURA brings about 65% fewer storage costs for entity embeddings when 70% of the entries are set to 0. Therefore, DURA can significantly reduce the storage usage while maintaining satisfying performance.

6 Conclusion

We propose a widely applicable and effective regularizer—namely, DURA—for tensor factorization based knowledge graph completion models. DURA is based on the observation that, for an existing tensor factorization based KGC model (primal), there is often another distance based KGC model (dual) closely associated with it. Experiments show that DURA brings consistent and significant improvements to TFB models on benchmark datasets. Moreover, visualization results show that DURA can encourage entities with similar semantics to have similar embeddings, which is beneficial to the prediction of unknown triplets. Since the current formulation of DURA is designed for tensor factorization based models, a direction for future work is to extend DURA to models in other categories, such as KBAT [21] and GAATs [32].
Broader Impact

The proposed regularizer DURA can significantly improve the performance of tensor factorization based knowledge graph completion models. In other words, it can help us to predict missing links in knowledge graphs automatically. Therefore, using models with DURA, we do not need to complete knowledge graphs manually. A great amount of manpower can be saved, and work efficiency can be increased. After the completion process, knowledge graphs can provide volume and valuable human knowledge in a structured way. They can be applied to many scenarios that require human knowledge. For example, an E-commerce company can use knowledge graphs for customer service and personalized recommendation. Medical workers can use them to make a diagnosis.

One ethical concern when using automatic knowledge graph completion methods is the potential for privacy disclosure. If we use public data on the Internet to construct a knowledge graph and then complete it using the proposed method, personal information that one does not want to make public may be unveiled. Therefore, we advise everyone to be cautious about usage scenarios of automatic knowledge graph completion methods.

Acknowledgments and Disclosure of Funding

We would like to thank all the anonymous reviewers for their insightful comments. This work was supported in part by NSFC (61822604, 61836006, U19B2026).

References


