HAL open science

Canonical Tensor Decomposition for Knowledge Base Completion

Timothée Lacroix, Nicolas Usunier, Guillaume Obozinski

To cite this version:

Timothée Lacroix, Nicolas Usunier, Guillaume Obozinski. Canonical Tensor Decomposition for Knowledge Base Completion. Proceedings of the 35th International Conference on Machine Learning, 2018, 80, pp.2863-2872. 10.48550/arXiv.1806.07297 . hal-01817595

HAL Id: hal-01817595

https://enpc.hal.science/hal-01817595

Submitted on 18 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Canonical Tensor Decomposition for Knowledge Base Completion

Timothée Lacroix [1 2] **Nicolas Usunier** [1] **Guillaume Obozinski** [2]

Abstract

The problem of Knowledge Base Completion can be framed as a 3rd-order binary tensor completion problem. In this light, the Canonical Tensor Decomposition (CP) (Hitchcock, 1927) seems like a natural solution; however, current implementations of CP on standard Knowledge Base Completion benchmarks are lagging behind their competitors. In this work, we attempt to understand the limits of CP for knowledge base completion. First, we motivate and test a novel regularizer, based on tensor nuclear p-norms. Then, we present a reformulation of the problem that makes it invariant to arbitrary choices in the inclusion of predicates or their reciprocals in the dataset. These two methods combined allow us to beat the current state of the art on several datasets with a CP decomposition, and obtain even better results using the more advanced ComplEx model.

1. Introduction

In knowledge base completion, the learner is given triples (subject, predicate, object) of facts about the world, and has to infer new triples that are likely but not yet known to be true. This problem has attracted a lot of attention (Nickel et al., 2016a; Nguyen, 2017) both as an example application of large-scale tensor factorization, and as a benchmark of learning representations of relational data.

The standard completion task is link prediction, which consists in answering queries (subject, predicate, ?) or (?, predicate, object). In that context, the canonical decomposition of tensors (also called CANDECOMP/PARAFAC or CP) (Hitchcock, 1927) is known to perform poorly compared to more specialized methods. For instance, DistMult (Yang et al., 2014), a particular case of CP which shares the factors for the subject and object modes, was recently shown to have state-of-the-art results (Kadlec et al., 2017). This result is surprising because DistMult learns a tensor that is symmetric in the subject and object modes, while the datasets contain mostly non-symmetric predicates.

The goal of this paper is to study whether and how CP can perform as well as its competitors. To that end, we evaluate three possibilities.

First, as Kadlec et al. (2017) showed that performances for these tasks are sensitive to the loss function and optimization parameters, we re-evaluate CP with a broader parameter search and a multiclass log-loss.

Second, since the best performing approaches are less expressive than CP, we evaluate whether regularization helps. On this subject, we show that the standard regularization used in knowledge base completion does not correspond to regularization with a tensor norm. We then propose to use tensor nuclear p-norms (Friedland & Lim, 2018), with the goal of designing more principled regularizers.

Third, we propose a different formulation of the objective, in which we model separately predicates and their inverse: for each predicate pred, we create an inverse predicate pred^{-1} and create a triple $(\mathrm{obj}, \mathrm{pred}^{-1}, \mathrm{sub})$ for each training triple $(\mathrm{sub}, \mathrm{pred}, \mathrm{obj})$. At test time, queries of the form $(?, \mathrm{pred}, \mathrm{obj})$ are answered as $(\mathrm{obj}, \mathrm{pred}^{-1}, ?)$. Similar formulations were previously used by Shen et al. (2016) and Joulin et al. (2017), but for different models for which there was no clear alternative, so the impact of this reformulation has never been evaluated.

To assess whether the results we obtain are specific to CP, we also carry on the same experiments with a state-of-the-art model, ComplEx (Trouillon et al., 2016). ComplEx has the same expressivity as CP in the sense that it can represent any tensor, but it implements a specific form of parameter sharing. We perform all our experiments on 5 common benchmark datasets of link prediction in knowledge bases.

Our results first confirm that within a reasonable time budget, the performance of both CP and ComplEx are highly dependent on optimization parameters. With systematic parameter searches, we obtain better results for ComplEx than what was previously reported, confirming its status as a state-of-the-art model on all datasets. For CP, the results

[1]Facebook AI Research, Paris, France [2]Université Paris-Est, Equipe Imagine, LIGM (UMR8049) Ecole des Ponts ParisTech Marne-la-Vallée, France. Correspondence to: Lacroix Timothee <timothee.lax@gmail.com>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

are still way below its competitors.

Learning and predicting with the inverse predicates, however, changes the picture entirely. First, with both CP and ComplEx, we obtain significant gains in performance on all the datasets. More precisely, we obtain state-of-the-art results with CP, matching those of ComplEx. For instance, on the benchmark dataset FB15K (Bordes et al., 2013), the mean reciprocal rank of vanilla CP and vanilla ComplEx are 0.40 and 0.80 respectively, and it grows to 0.86 for both approaches when modeling the inverse predicates.

Finally, the new regularizer we propose based on the nuclear 3-norm, does not dramatically help CP, which leads us to believe that a careful choice of regularization is not crucial for these CP models. Yet, for both CP and ComplEx with inverse predicates, it yields small but significant improvements on the more difficult datasets.

2. Tensor Factorization of Knowledge Bases

We describe in this section the formal framework we consider for knowledge base completion and more generally link prediction in relational data, the learning criteria, as well as the approaches that we will discuss.

2.1. Link Prediction in Relational Data

We consider relational data that comes in the form of triples (subject, predicate, object), where the subject and the object are from the same set of entities. In knowledge bases, these triples represent facts about entities of the world, such as $(Washington, capital_of, USA)$. A training set $\mathcal{S}$ contains triples of indices $\mathcal{S} = \{(i_1, j_1, k_1), ..., (i_{|\mathcal{S}|}, j_{|\mathcal{S}|}, k_{|\mathcal{S}|})\}$ that represent predicates that are known to hold. The validation and test sets contain queries of the form $(?, j, k)$ and $(i, j, ?)$, created from triples (i, j, k) that are known to hold but held-out from the training set. To give orders of magnitude, the largest datasets we experiment on, FB15K and YAGO3-10, contain respectively $15k/1.3k$ and $123k/37$ entities/predicates.

2.2. Tensor Decomposition for Link Prediction

Relational data can be represented as a $\{0, 1\}$-valued third order tensor $\boldsymbol{Y} \in \{0, 1\}^{N \times P \times N}$, where N is the total number of entities and P the number of predicates, with $\boldsymbol{Y}_{i,j,k} = 1$ if the relation (i, j, k) is known. In the rest of the paper, the three modes will be called the subject mode, the predicate mode and the object mode respectively. Tensor factorization algorithms can thus be used to infer a predicted tensor $\hat{\boldsymbol{X}} \in \mathbb{R}^{N \times P \times N}$ that approximates $\boldsymbol{Y}$ in a sense that we describe in the next subsection. Validation/test queries $(?, j, k)$ are answered by ordering entities i' by decreasing values of $\hat{\boldsymbol{X}}_{i',j,k}$, whereas queries $(i, j, ?)$ are answered by ordering entities k' by decreasing values of $\hat{\boldsymbol{X}}_{i,j,k'}$.

Several approaches have considered link prediction as a low-rank tensor decomposition problem. These models then differ only by structural constraints on the learned tensor. Three models of interest are:

CP. The canonical decomposition of tensors, also called CANDECOM/PARAFAC (Hitchcock, 1927), represents a tensor $\boldsymbol{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ as a sum of R rank one tensors $u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(3)}$ (with $\otimes$ the tensor product) where $r \in \{1, ..., R\}$, and $u_r^{(m)} \in \mathbb{R}^{N_m}$:

$$\boldsymbol{X} = \sum_{r=1}^{R} u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(3)} \,.$$

A representation of this decomposition, and the score of a specific triple is given in Figure 1 (a). Given $\boldsymbol{X}$, the smallest R for which this decomposition holds is called the canonical rank of $\boldsymbol{X}$.

DistMult. In the more specific context of link prediction, it has been suggested in Bordes et al. (2011); Nickel et al. (2011) that since both subject and object mode represent the same entities, they should have the same factors. DistMult (Yang et al., 2014) is a version of CP with this additional constraint. It represents a tensor $\boldsymbol{X} \in \mathbb{R}^{N \times P \times N}$ as a sum of rank-1 tensors $u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(1)}$:

$$\boldsymbol{X} = \sum_{r=1}^{R} u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(1)} \,.$$

ComplEx. By contrast with the first models that proposed to share the subject and object mode factors, DistMult yields a tensor that is symmetric in the object and subject modes. The assumption that the data tensor can be properly approximated by a symmetric tensor for Knowledge base completion is not satisfied in many practical cases (e.g., while $(Washington, capital_of, USA)$ holds, $(USA, capital_of, Washington)$ does not). ComplEx (Trouillon et al., 2016) proposes an alternative where the subject and object modes share the parameters of the factors, but are complex conjugate of each other. More precisely, this approach represents a real-valued tensor $\boldsymbol{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ as the real part of a sum of R complex-valued rank one tensors $u_r^{(1)} \otimes u_r^{(2)} \otimes \overline{u}_r^{(1)}$ where $r \in \{1, ..., R\}$, and $u_r^{(m)} \in \mathbb{C}^{N_m}$

$$\boldsymbol{X} = \mathrm{Re}\big(\sum_{r=1}^{R} u_r^{(1)} \otimes u_r^{(2)} \otimes \overline{u}_r^{(1)}\big) \,,$$

where $\overline{u}_r^{(1)}$ is the complex conjugate of $u_r^{(1)}$. This decomposition can represent any real tensor (Trouillon et al., 2016).

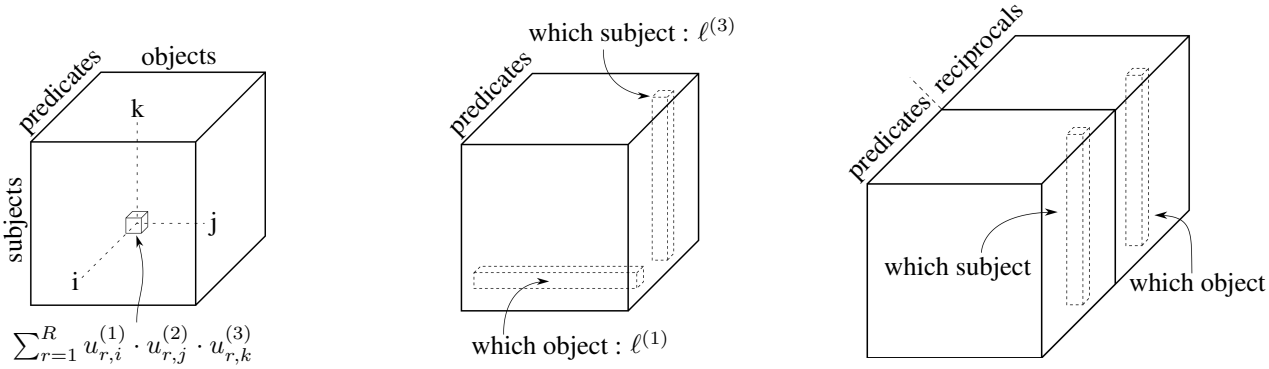

Figure 1. (a) On the left, the link between the score of a triple (i,j,k) and the tensor estimated via CP. (b) In the middle, the two type of fiber losses that we will consider. (c) On the right, our semantically invariant reformulation, the first-mode fibers become third-mode fibers of the reciprocal half of the tensor.

The good performances of DistMult on notoriously non-symmetric datasets such as FB15K or WN18 are surprising. First, let us note that for the symmetricity to become an issue, one would have to evaluate queries $(i, j, ?)$ while also trying to answer correctly to queries of the form $(?, j, i)$ for a non-symmetric predicate j. The ranking for these two queries would be identical, and thus, we can expect issues with relations such as $capital_of$. In FB15K, those type of problematic queries make up only 4% of the test set and thus, have a small impact. On WN18 however, they make up 60% of the test set. We describe in appendix 8.1 a simple strategy for DistMult to have a high filtered MRR on the hierarchical predicates of WN18 despite its symmetricity assumption.

2.3. Training

Previous work suggested ranking losses (Bordes et al., 2013), binary logistic regression (Trouillon et al., 2016) or sampled multiclass log-loss (Kadlec et al., 2017). Motivated by the solid results in Joulin et al. (2017), our own experimental results, and with a satisfactory speed of about two minutes per epoch on FB15K, we decided to use the full multiclass log-loss.

Given a training triple (i, j, k) and a predicted tensor $\boldsymbol{X}$, the instantaneous multi-class log-loss $\ell_{i,j,k}(\boldsymbol{X})$ is

$$\begin{aligned} \ell_{i,j,k}(\boldsymbol{X}) &= \ell^{(1)}_{i,j,k}(\boldsymbol{X}) + \ell^{(3)}_{i,j,k}(\boldsymbol{X}) \qquad (1) \\ \ell^{(1)}_{i,j,k}(\boldsymbol{X}) &= -\boldsymbol{X}_{i,j,k} + \log\big(\sum_{k'} \exp(\boldsymbol{X}_{i,j,k'})\big) \\ \ell^{(3)}_{i,j,k}(\boldsymbol{X}) &= -\boldsymbol{X}_{i,j,k} + \log\big(\sum_{i'} \exp(\boldsymbol{X}_{i',j,k})\big)\,. \end{aligned}$$

These two partial losses are represented in Figure 1 (b). For CP, the final tensor is computed by finding a minimizer of a regularized empirical risk formulation, where the factors $u_r^{(d)}$ are weighted in a data-dependent manner by $w_{\mathcal{S}}^{(d)}$, which we describe below:

$$\min_{\substack{(u_r^{(d)})_{d=1..3} \\ r=1..R}} \sum_{(i,j,k)\in\mathcal{S}} \ell_{i,j,k}\Big(\sum_{r=1}^{R} u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(3)}\Big) + \lambda \sum_{r=1}^{R}\sum_{d=1}^{3} \|w_{\mathcal{S}}^{(d)} \odot u_r^{(d)}\|_2^2\,, \qquad (2)$$

where $\odot$ is the entry-wise multiplication of vectors. For DistMult and ComplEx, the learning objective is similar, up to the appropriate parameter sharing and computation of the tensor.

As discussed in Section 3.2, the weights $w_{\mathcal{S}}^{(d)}$ may improve performances when some rows/columns are sampled more than others. They appear naturally in optimization with stochastic gradient descent when the regularizer is applied only to the parameters that are involved in the computation of the instantaneous loss. For instance, in the case of the logistic loss with negative sampling used by Trouillon et al. (2016), denoting by q_i^d the marginal probability (over $\mathcal{S}$) that index i appears in mode d of a data triple, these weights are $w_{\mathcal{S},i}^{(d)} = \sqrt{q_i^d + \alpha}$ for some $\alpha > 0$ that depends on the negative sampling scheme.

We focus on redefining the loss (1) and the regularizer (2).

3. Related Work

We discuss here in more details the work that has been done on link prediction in relational data and on regularizers for tensor completion.

3.1. Link Prediction in Relational Data

There has been extensive research on link prediction in relational data, especially in knowledge bases, and we review here only the prior work that is most relevant to this paper. While some approaches explicitly use the graph struc-

ture during inference (Lao et al., 2011), we focus here on representation learning and tensor factorization methods, which are the state-of-the-art on the benchmark datasets we use. We also restrict the discussion to approaches that only use relational information, even though some approaches have been proposed to leverage additional types (Krompass et al., 2015; Ma et al., 2017) or external word embeddings (Toutanova & Chen, 2015).

We can divide the first type of approaches into two broad categories. First, two-way approaches score a triple (i, j, k) depending only on bigram interaction terms of the form subject-object, subject-predicate, and predicate-object. Even though they are tensor approximation algorithms of limited expressivity, two-way models based on translations TransE, or on bag-of-word representations (Joulin et al., 2017) have proved competitive on many benchmarks. Yet, methods using three-way multiplicative interactions, as described in the previous section, show the strongest performances (Bordes et al., 2011; Garcia-Duran et al., 2016; Nickel et al., 2016b; Trouillon et al., 2016). Compared to general-purpose tensor factorization methods such as CP, a common feature of these approaches is to share parameters between objects and subjects modes (Nickel et al., 2011), an idea that has been widely accepted except for the two-way model of Joulin et al. (2017). DistMult (Yang et al., 2014) is the extreme case of this parameter sharing, in which the predicted tensor is symmetric in the subject and object modes.

3.2. Regularization for Matrix Completion

Norm-based regularization has been extensively studied in the context of matrix completion. The trace norm (or nuclear norm) has been proposed as a convex relaxation of the rank (Srebro et al., 2005) for matrix completion in the setting of rating prediction, with strong theoretical guarantees (Candès & Recht, 2009). While efficient algorithms to solve the convex problems have been proposed (see e.g. Cai et al., 2010; Jaggi et al., 2010), the practice is still to use the matrix equivalent of the nonconvex formulation (2). For the trace norm (nuclear 2-norm), in the matrix case, the regularizer simply becomes the squared 2-norm of the factors and lends itself to alternating methods or SGD optimization (Rennie & Srebro, 2005; Koren et al., 2009). When the samples are not taken uniformly at random from a matrix, some other norms are preferable to the usual nuclear norm. The weighted trace norm reweights elements of the factors based on the marginal rows and columns sampling probabilities, which can improve sample complexity bounds when sampling is non-uniform (Foygel et al., 2011; Negahban & Wainwright, 2012). Direct SGD implementations on the nonconvex formulation implicitly take this reweighting rule into account and were used by the winners of the Netflix challenge (see Srebro & Salakhutdinov, 2010, Section 5).

3.3. Tensor Completion and Decompositions

There is a large body of literature on low-rank tensor decompositions (see Kolda & Bader, 2009, for a comprehensive review). Closely related to our work is the canonical decomposition of tensor (also called CANDECOMP/PARAFAC or CP) (Hitchcock, 1927), which solves a problem similar to (4) without the regularization (i.e., $\lambda = 0$), and usually the square loss.

Several norm-based regularizations for tensors have been proposed. Some are based on unfolding a tensor along each of its modes to obtain matricizations, and either regularize by the sum of trace norms of the matricizations (Tomioka et al., 2010) or write the original tensor as a sum of tensors T_k, regularizing their respective kth matricizations with the trace norm (Wimalawarne et al., 2014). However, in the large-scale setting, even rank-1 approximations of matricizations involve too many parameters to be tractable.

Recently, the tensor trace norm (nuclear 2-norm) was proposed as a regularizer for tensor completion Yuan & Zhang (2016), and an algorithm based on the generalized conditional gradient has been developed by Cheng et al. (2016). This algorithm requires, in an inner loop, to compute a (constrained) rank-1 tensor that has largest dot-product with the gradient of the data-fitting term (gradient w.r.t. the tensor argument). This algorithm is efficient in our setup only with the square error loss (instead of the multiclass log-loss), because the gradient is then a low-rank + sparse tensor when the argument is low-rank. However, on large-scale knowledge bases, the state of the art is to use a binary log-loss or a multiclass log-loss (Trouillon et al., 2016; Kadlec et al., 2017); in that case, the gradient is not adequately structured, thereby causing the approach of (Cheng et al., 2016) to be too computationally costly.

4. Nuclear p-Norm Regularization

As discussed in Section 3, norm-based regularizers have proved useful for matrices. We aim to reproduce these successes with tensor norms. We use the nuclear p-norms defined by Friedland & Lim (2018). As shown in Equation (2), the community has favored so far a regularizer based on the square Frobenius norms of the factors (Yang et al., 2014; Trouillon et al., 2016). We first show that the unweighted version of this regularizer is not a tensor norm. Then, we propose4 a variational form of the nuclear 3-norm to replace the usual regularization at no additional computational cost when used with SGD. Finally, we discuss a weighting scheme analogous to the weighted trace-norm proposed in Srebro & Salakhutdinov (2010).

4.1. From Matrix Trace-Norm to Tensor Nuclear Norms

To simplify notation, let us introduce the set of CP decompositions of a tensor $\boldsymbol{X}$ of rank at most R:

$$\mathcal{U}_R(\boldsymbol{X}) = \Big\{ (u_r^{(d)})_{\substack{d=1..3\\ r=1..R}} \;\Big|\; \boldsymbol{X} = \sum_{r=1}^{R} u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(3)}, \quad \forall r, d,\ u_r^{(d)} \in \mathbb{R}^{N_d} \Big\}.$$

We will study the family of regularizers:

$$\Omega_p^{\alpha}(u) = \frac{1}{3} \sum_{r=1}^{R} \sum_{d=1}^{3} \|u_r^{(d)}\|_p^{\alpha}.$$

Note that with $p = \alpha = 2$, we recover the familiar squared Frobenius norm regularizer used in (2). Similar to showing that the squared Frobenius norm is a *variational form* of the trace norm on matrices (i.e., its minimizers realize the trace norm, $\inf_{M=UV^T} \frac{1}{2}(\|U\|_F^2 + \|V\|_F^2) = \|M\|_*$), we start with a technical lemma that links our regularizer with a function on the spectrum of our decompositions.

Lemma 1.

$$\min_{u \in \mathcal{U}_R(\boldsymbol{X})} \frac{1}{3} \sum_{r=1}^{R} \sum_{d=1}^{3} \|u_r^{(d)}\|_p^{\alpha} = \min_{u \in \mathcal{U}_R(\boldsymbol{X})} \sum_{r=1}^{R} \prod_{d=1}^{3} \|u_r^{(d)}\|_p^{\alpha/3}.$$

Moreover, the minimizers of the left-hand side satisfy:

$$\|u_r^{(d)}\|_p = \sqrt[3]{\prod_{d'=1}^{3} \|u_r^{(d')}\|_p}.$$

Proof. See Appendix 8.2. □

This Lemma motivates the introduction of the set of p-norm normalized tensor decompositions:

$$\overline{\mathcal{U}}_R^p(\boldsymbol{X}) = \Big\{ (\sigma_r, (\tilde{u}_r))_{r=1..R} \;\Big|\; \sigma_r = \prod_{d=1}^{3} \|u_r^{(d)}\|_p, \quad \tilde{u}_r^{(d)} = \frac{u_r^{(d)}}{\|u_r^{(d)}\|_p}, \forall r, d, u \in \mathcal{U}_R(\boldsymbol{X}) \Big\}.$$

Lemma 2, shows that Ω_p^{α} behaves as an $\ell_{\alpha/D}$ penalty over the CP *spectrum* for tensors of order D. We recover the nuclear norm for matrices when $\alpha = p = 2$.

Using Lemma 2, we have :

$$\min_{u \in \mathcal{U}_R(\boldsymbol{X})} \Omega_2^2(u) \leq \eta \iff \min_{(\sigma, \tilde{u}) \in \overline{\mathcal{U}}_R^2(\boldsymbol{X})} \|\sigma\|_{2/3} \leq \eta^{3/2} \tag{3}$$

We show that the sub-level sets of the term on the right are not convex, which implies that Ω_2^2 is not the variational form of a tensor norm, and hence, is not the tensor analog to the matrix trace norm.

Proposition 1. *The function over third order-tensors of* $\mathbb{R}^{N_1 \times N_2 \times N_3}$ *defined as*

$$|||\boldsymbol{X}||| = \min \Big\{ \|\sigma\|_{2/3} \;\Big|\; (\sigma, \tilde{u}) \in \overline{\mathcal{U}}_R^2(\boldsymbol{X}), R \in \mathbb{N} \Big\}$$

is not convex.

Proof. See Appendix 8.2. □

Remark 1. *Cheng et al. (2016, Appendix D) already showed that regularizing with the square Frobenius norm of the factors is not related to the trace norm for tensors of order 3 and above, but their observation is that the regularizer is not positively homogeneous, i.e.,* $\min_{u \in \alpha \mathcal{U}_R(\boldsymbol{X})} \Omega_2^2(u) \neq |\alpha| \min_{u \in \mathcal{U}_R(\boldsymbol{X})} \Omega_2^2(u)$. *Our result in Proposition 1 is stronger in that we show that this regularizer is not a norm even after the rescaling (3) to make it homogeneous.*

The nuclear p-norm of $\boldsymbol{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ for $p \in [1, +\infty]$, is defined in Friedland & Lim (2018) as

$$\|\boldsymbol{X}\|_{*,p} := \min \Big\{ \|\sigma\|_1 \;\Big|\; (\sigma, \tilde{u}) \in \overline{\mathcal{U}}_R^p(\boldsymbol{X}), R \in \mathbb{N} \Big\}.$$

Given an estimated upper bound on the optimal R, the original problem (2) can then be re-written as a non-convex problem using the equivalence in Lemma 2:

$$\min_{\substack{(u_r^{(d)})_{\substack{d=1..3\\ r=1..R}}}} \sum_{(i,j,k) \in \mathcal{S}} \ell_{i,j,k}\Big(\sum_{r=1}^{R} u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(3)} \Big) + \frac{\lambda}{3} \sum_{r=1}^{R} \sum_{d=1}^{3} \|u_r^{(d)}\|_p^3. \tag{4}$$

This variational form suggests to use $p = 3$, as a means to make the regularizer separable in each coefficients, given that then $\|u_r^{(d)}\|_p^3 = \sum_{i=1}^{n_d} |u_{r,i}^{(d)}|^3$.

4.2. Weighted Nuclear p-Norm

Similar to the weighted trace-norm for matrices, the weighted nuclear 3-norm can be easily implemented by keeping the regularization terms corresponding to the sampled triplets only, as discussed in Section 3.2. This leads to a formulation of the form

$$\min_{\substack{(u_r^{(d)})_{\substack{d=1..3\\ r=1..R}}}} \sum_{(i,j,k) \in \mathcal{S}} \Big[\ell_{i,j,k}\big(\sum_{r=1}^{R} u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(3)} \big) + \frac{\lambda}{3} \sum_{r=1}^{R} \Big(|u_{r,i}^{(1)}|^3 + |u_{r,j}^{(2)}|^3 + |u_{r,k}^{(3)}|^3 \Big) \Big]. \tag{5}$$

For an example (i, j, k), only the parameters involved in the computation of $\hat{X}_{i,j,k}$ are regularized. The computational complexity is thus the same as the currently used weighted

Frobenius norm regularizer. With $q^{(1)}$ (resp. $q^{(2)}$, $q^{(3)}$) the marginal probabilities of sampling a subject (resp. predicate, object), the weighting implied by this regularization scheme is

$$\|\boldsymbol{X}\|_{*,3,w} = \|(\sqrt[3]{q^{(1)}} \otimes \sqrt[3]{q^{(2)}} \otimes \sqrt[3]{q^{(3)}}) \odot \boldsymbol{X}\|_{*,3}$$

We justify this weighting only by analogy with the matrix case discussed by (Srebro & Salakhutdinov, 2010): to make the weighted nuclear 3-norm of the all 1 tensor independent of its dimensions for a uniform sampling (since the nuclear 3-norm grows as $\sqrt[3]{MNP}$ for an (M, N, P) tensor).

Comparatively, for the weighted version of the nuclear 2-norm analyzed in Yuan & Zhang (2016), the nuclear 2-norm of the all 1 tensor scales like $\sqrt{NMP}$. This would imply a formulation of the form

$$\min_{\substack{(u_r^{(d)})_{\substack{d=1..3\\r=1..R}}}} \sum_{(i,j,k)\in\mathcal{S}} \ell_{i,j,k}\Big(\sum_{r=1}^{R} u_r^{(1)} \otimes u_r^{(2)} \otimes u_r^{(3)}\Big) + \frac{\lambda}{3}\sum_{r=1}^{R}\sum_{d=1}^{3}\|\sqrt{q^{(d)}} \odot u_r^{(d)}\|_2^3 \,. \tag{6}$$

Contrary to formulation (5), the optimization of formulation (6) with a minibatch SGD leads to an update of every coefficients for each mini-batch considered. Depending on the implementation, and size of the factors, there might be a large difference in speed between the updates of the weighted nuclear $\{2, 3\}$-norm. In our implementation, this difference for CP is of about $1.5\times$ in favor of the nuclear 3-norm on FB15K.

5. A New CP Objective

Since our evaluation objective is to rank either the left-hand side or right-hand side of the predicates in our dataset, what we are trying to achieve is to model both predicates and their reciprocal. This suggests appending to our input the reciprocals of each predicates, thus factorizing $[\boldsymbol{Y};_2 \tilde{\boldsymbol{Y}}]$ rather than $\boldsymbol{Y}$, where $[\ ;_2\ ]$ is the mode-2 concatenation, and $\boldsymbol{Y}_{i,j,k} = \tilde{\boldsymbol{Y}}_{k,j,i}$. After that, we only need to model the object fibers of this new tensor $\boldsymbol{Y}$. We represent this transformation in Figure 1 (c). This reformulation has an important side-effect: it makes our algorithm invariant to the arbitrary choice of including a predicate or its reciprocal in the dataset. This property was introduced as "Semantic Invariance" in Bailly et al. (2015). Another way of achieving this invariance property would be to find the flipping of predicates that lead to the smallest model. In the case of a CP decomposition, we would try to find the flipping that leads to lowest tensor rank. This seems hopeless, given the NP-hardness of computing the tensor rank.

More precisely, the instantaneous loss of a training triple (i, j, k) becomes :

$$\ell_{i,j,k}(\boldsymbol{X}) = -\boldsymbol{X}_{i,j,k} + \log\big(\sum_{k'}\exp(\boldsymbol{X}_{i,j,k'})\big) - \boldsymbol{X}_{k,j+P,i} + \log\big(\sum_{i'}\exp(\boldsymbol{X}_{k,j+P,i'})\big). \tag{7}$$

At test time we use $\hat{\boldsymbol{X}}_{i,j,:}$ to rank possible right hand sides for query $(i, j, ?)$ and $\hat{\boldsymbol{X}}_{k,j+P,:}$ to rank possible left hand sides for query $(?, j, k)$.

Using CP to factor the tensor described in (7), we beat the previous state of the art on many benchmarks, as shown in Table 2. This reformulation seems to help even the ComplEx decomposition, for which parameters are shared between the entity embeddings of the first and third mode.

Dataset	N	P	Train	Valid	Test
WN18	41k	18	141k	5k	5k
WN18RR	41k	11	87k	3k	3k
FB15K	15k	1k	500k	50k	60k
FB15K-237	15k	237	272k	18k	20k
YAGO3-10	123k	37	1M	5k	5k

Table 1. Dataset statistics.

6. Experiments

We conducted all experiments on a Quadro GP 100 GPU. The code is available at `https://github.com/facebookresearch/kbc`.

6.1. Datasets and Experimental Setup

WN18 and FB15K are popular benchmarks in the Knowledge Base Completion community. The former comes from the WordNet database, was introduced in Bordes et al. (2014) and describes relations between words. The most frequent types of relations are highly hierarchical (e.g., hypernym, hyponym). The latter is a subsampling of Freebase limited to 15k entities, introduced in Bordes et al. (2013). It contains predicates with different characteristics (e.g., one-to-one relations such as *capital_of* to many-to-many such as *actor_in_film*).

Toutanova & Chen (2015) and Dettmers et al. (2017) identified train to test leakage in both these datasets, in the form of test triplets, present in the train set for the reciprocal predicates. Thus, both of these authors created two modified datasets: FB15K-237 and WN18RR. These datasets are harder to fit, so we expect regularization to have more impact. Dettmers et al. (2017) also introduced the dataset YAGO3-10, which is larger in scale and doesn't suffer from leakage. All datasets statistics are shown in Table 1.

	Model	WN18		WN18RR		FB15K		FB15K-237		YAGO3-10	
		MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10
Past SOTA	CP	0.08	0.13	-	-	0.33	0.53	-	-	-	-
	ComplEx †	0.94	0.95	0.44	0.51	0.70	0.84	0.25	0.43	0.36	0.55
	DistMult *	0.82	0.94	0.43	0.49	0.80	0.89	0.24	0.42	0.34	0.54
	ConvE *	0.94	0.95	0.46	0.48	0.75	0.87	0.32	0.49	0.52	0.66
	Best Published⋆	0.94	**0.97**	0.46	0.51	0.84	**0.93**	0.32	0.49	0.52	0.66
Standard	CP-N3	0.20	0.33	0.12	0.20	0.46	0.65	0.33	0.51	0.38	0.65
	ComplEx-N3	**0.95**	**0.96**	0.47	0.54	0.80	0.89	0.35	0.54	0.49	0.68
Reciprocal	CP-FRO	**0.95**	0.95	0.46	0.48	**0.86**	0.91	0.34	0.51	0.54	0.68
	CP-N3	**0.95**	0.96	0.47	0.54	**0.86**	0.91	0.36	0.54	0.57	**0.71**
	ComplEx-FRO	**0.95**	0.96	0.47	0.54	**0.86**	0.91	0.35	0.53	0.57	**0.71**
	ComplEx-N3	**0.95**	0.96	**0.48**	**0.57**	**0.86**	0.91	**0.37**	**0.56**	**0.58**	**0.71**

Table 2. *Results taken as best from Dettmers et al. (2017) and Kadlec et al. (2017). †Results taken as best from Dettmers et al. (2017) and Trouillon et al. (2016).⋆ We give the origin of each result on the Best Published row in appendix.

In all our experiments, we distinguish two settings: Reciprocal, in which we use the loss described in equation (7) and Standard, which uses the loss in equation (1). We compare our implementation of CP and ComplEx with the best published results, then the different performances between the two settings, and finally, the contribution of the regularizer in the reciprocal setting. In the Reciprocal setting, we compare the weighted nuclear 3-norm (N3) against the regularizer described in (2) (FRO). In preliminary experiments, the weighted nuclear 2-norm described in (6) did not seem to perform better than N3 and was slightly slower. We used Adagrad (Duchi et al., 2011) as our optimizer, whereas Kadlec et al. (2017) favored Adam (Kingma & Ba, 2014), because preliminary experiments didn't show improvements.

We ran the same grid for all algorithms and regularizers on the FB15K, FB15K-237, WN18, WN18RR datasets, with a rank set to 2000 for ComplEx, and 4000 for CP. Our grid consisted of two learning rates: 10^{-1} and 10^{-2}, two batch-sizes: 25 and 100, and regularization coefficients in $[0, 10^{-3}, 5.10^{-3}, 10^{-2}, 5.10^{-2}, 10^{-1}, 5.10^{-1}]$. On YAGO3-10, we limited our models to rank 1000 and used batch-sizes 500 and 3000, the rest of the grid was identical. We used the train/valid/test splits provided with these datasets and measured the filtered Mean Reciprocal Rank (MRR) and Hits@10 (Bordes et al. (2013)). We used the filtered MRR on the validation set for early stopping and report the corresponding test metrics. In this setting, an epoch for ComplEx with batch-size 100 on FB15K took about $110s$ and $325s$ for a batch-size of 25. We trained for 100 epochs to ensure convergence, reported performances were reached within the first 25 epochs.

All our results are reported in Table 2 and will be discussed in the next subsections. Besides our implementations of CP and ComplEx, we include the results of ConvE and DistMult in the baselines. The former because Dettmers et al. (2017) includes performances on the WN18RR and YAGO3-10 benchmarks, the latter because of the good performances on FB15K of DistMult and the extensive experiments on WN18 and FB15K reported in Kadlec et al. (2017). The performances of DistMult on FB15K-237, WN18RR and YAGO3-10 may be slightly underestimated, since our baseline CP results are better. To avoid clutter, we did not include in our table of results algorithms that make use of external data such as types (Krompass et al., 2015), external word embeddings (Toutanova & Chen, 2015), or using path queries as regularizers (Guu et al., 2015). The published results corresponding to these methods are subsumed in the "Best Published" line of Table 2, which is taken, for every single metric and dataset, as the best published result we were able to find.

6.2. Reimplementation of the Baselines

The performances of our reimplementation of CP and ComplEx appear in the middle rows of Table 2 (Standard setting). We only kept the results for the nuclear 3-norm, which didn't seem to differ from those with the Frobenius norm. Our results are slightly better than their published counterparts, going from 0.33 to 0.46 filtered MRR on FB15K for CP and 0.70 to 0.80 for ComplEx. This might be explained in part by the fact that in the Standard setting (2) we use a multi-class log-loss, whereas Trouillon et al. (2016) used binomial negative log-likelihood. Another reason for this increase can be the large rank of 2000 that we chose, where

	1-1	m-1	1-m	m-m
CP Standard	0.45	0.71	0.24	0.44
CP Reciprocal	0.77	**0.92**	**0.71**	0.86
ComplEx Standard	0.87	**0.92**	0.59	0.81
ComplEx Reciprocal	**0.88**	**0.92**	**0.71**	**0.87**

Table 3. Average MRR per relation type on FB15K.

previously published results used a rank of around 200; the more extensive search for optimization/regularization parameters and the use of nuclear 3-norm instead of the usual regularization are also most likely part of the explanation.

6.3. Standard vs Reciprocal

In this section, we compare the effect of reformulation (7), that is, the middle and bottom rows of Table 2. The largest differences are obtained for CP, which becomes a state of the art contender going from 0.2 to 0.95 filtered MRR on WN18, or from 0.46 to 0.86 filtered MRR on FB15K.For ComplEx, we notice a weaker, but consistent improvement by using our reformulation, with the biggest improvements observed on FB15K and YAGO3-10. Following the analysis in Bordes et al. (2013), we show in Table 3 the average filtered MRR as a function of the degree of the predicates. We compute the average in and out degrees on the training set, and separate the predicates in 4 categories : 1-1, 1-m, m-1 and m-m, with a cut-off at 1.5 on the average degree. We include reciprocal predicates in these statistics. That is, a predicate with an average in-degree of 1.2 and average out-degree of 3.2 will count as a 1-m when we predict its right-hand side, and as an m-1 when we predict its left-hand side. Most of our improvements come from the 1-m and m-m categories, both on ComplEx and CP.

6.4. Frobenius vs Nuclear 3

We focus now on the effect of our norm-based N3 regularizer, compared to the Frobenius norm regularizer favored by the community. Comparing the four last rows of Table 2, we notice a small but consistent performance gain across datasets. The biggest improvements appear on the harder datasets WN18RR, FB15K-237 and YAGO3-10. We checked on WN18RR the significance of that gain with a Signed Rank test on the rank pairs for CP.

6.5. Effect of Optimization Parameters

During these experiments, we noticed a heavy influence of optimization hyper-parameters on final results. This influence can account for as much as 0.1 filtered MRR and is illustrated in Figure 2.

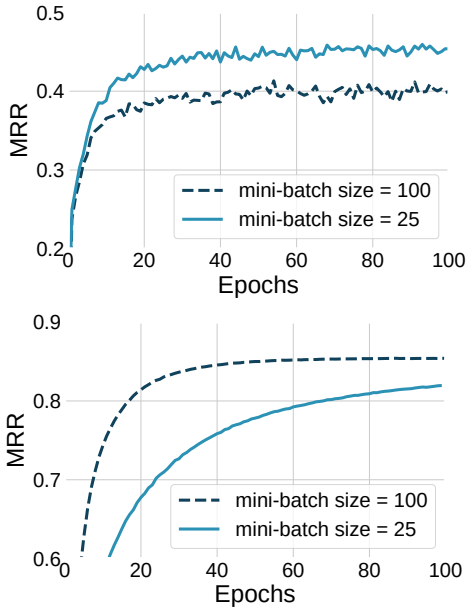

Figure 2. Effect of the batch-size on FB15K in the Standard (top) and Reciprocal (bottom) settings, other parameters being equal. The difference is large even after 100 epochs and the effect is inverted in the two settings, making it hard to choose the batch-size a priori.

7. Conclusion and Discussion

The main contribution of this paper is to isolate and systematically explore the effect of different factors for large-scale knowledge base completion. While the impact of optimization parameters was well known already, neither the effect of the formulation (adding reciprocals doubles the mean reciprocal rank on FB15K for CP) nor the impact of the regularization was properly assessed. The conclusion is that the CP model performs nearly as well as the competitors when each model is evaluated in its optimal configuration. We believe this observation is important to assess and prioritize directions for further research on the topic.

In addition, our proposal to use nuclear p-norm as regularizers with $p \neq 2$ for tensor factorization in general is of independent interest.

The results we present leave several questions open. Notably, whereas we give definite evidence that CP itself can perform extremely well on these datasets as long as the problem is formulated correctly, we do not have a strong theoretical justification as to why the differences in performances are so significant.

Acknowledgements

The authors thank Armand Joulin and Maximilian Nickel for valuable discussions.

References

Bailly, R., Bordes, A., and Usunier, N. Semantically Invariant Tensor Factorization. 2015.

Bordes, A., Weston, J., Collobert, R., and Bengio, Y. Learning structured embeddings of knowledge bases. In *Conference on artificial intelligence*, 2011.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 2787–2795. Curran Associates, Inc., 2013.

Bordes, A., Glorot, X., Weston, J., and Bengio, Y. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259, 2014.

Cai, J.-F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

Cheng, H., Yu, Y., Zhang, X., Xing, E., and Schuurmans, D. Scalable and sound low-rank tensor learning. In *Artificial Intelligence and Statistics*, pp. 1114–1123, 2016.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. Convolutional 2d knowledge graph embeddings. *arXiv preprint arXiv:1707.01476*, 2017.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Foygel, R., Shamir, O., Srebro, N., and Salakhutdinov, R. R. Learning with the weighted trace-norm under arbitrary sampling distributions. In *Advances in Neural Information Processing Systems*, pp. 2133–2141, 2011.

Friedland, S. and Lim, L.-H. Nuclear norm of higher-order tensors. *Mathematics of Computation*, 87(311):1255–1281, 2018.

Garcia-Duran, A., Bordes, A., Usunier, N., and Grandvalet, Y. Combining two and three-way embedding models for link prediction in knowledge bases. *Journal of Artificial Intelligence Research*, 55:715–742, 2016.

Guu, K., Miller, J., and Liang, P. Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 318–327, 2015.

Hitchcock, F. L. The expression of a tensor or a polyadic as a sum of products. *Studies in Applied Mathematics*, 6 (1-4):164–189, 1927.

Jaggi, M., Sulovsk, M., and others. A simple algorithm for nuclear norm regularized problems. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 471–478, 2010.

Joulin, A., Grave, E., Bojanowski, P., Nickel, M., and Mikolov, T. Fast linear model for knowledge graph embeddings. *arXiv preprint arXiv:1710.10881*, 2017.

Kadlec, R., Bajgar, O., and Kleindienst, J. Knowledge Base Completion: Baselines Strike Back. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 69–74, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

Krompass, D., Baier, S., and Tresp, V. Type-constrained representation learning in knowledge graphs. In *International Semantic Web Conference*, pp. 640–655. Springer, 2015.

Lao, N., Mitchell, T., and Cohen, W. W. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 529–539. Association for Computational Linguistics, 2011.

Ma, S., Ding, J., Jia, W., Wang, K., and Guo, M. Transt: Type-based multiple embedding representations for knowledge graph completion. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 717–733. Springer, 2017.

Negahban, S. and Wainwright, M. J. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13 (May):1665–1697, 2012.

Nguyen, D. Q. An overview of embedding models of entities and relationships for knowledge base completion. *arXiv preprint arXiv:1703.08098*, 2017.

Nickel, M., Tresp, V., and Kriegel, H.-P. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 809–816, 2011.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016a.

Nickel, M., Rosasco, L., Poggio, T. A., et al. Holographic embeddings of knowledge graphs. 2016b.

Rennie, J. D. and Srebro, N. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pp. 713–719. ACM, 2005.

Shen, Y., Huang, P.-S., Chang, M.-W., and Gao, J. Implicit reasonet: Modeling large-scale structured relationships with shared memory. 2016.

Srebro, N. and Salakhutdinov, R. R. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Advances in Neural Information Processing Systems*, pp. 2056–2064, 2010.

Srebro, N., Rennie, J., and Jaakkola, T. S. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pp. 1329–1336, 2005.

Tomioka, R., Hayashi, K., and Kashima, H. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.

Toutanova, K. and Chen, D. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66, 2015.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., and Bouchard, G. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pp. 2071–2080, 2016.

Wimalawarne, K., Sugiyama, M., and Tomioka, R. Multi-task learning meets tensor factorization: task imputation via convex optimization. In *Advances in neural information processing systems*, pp. 2825–2833, 2014.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Yuan, M. and Zhang, C.-H. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16(4):1031–1068, 2016.

8. Appendix

8.1. DistMult on hierarchical predicates

Suppose we are trying to embed a single hierarchical predicate p which is a n-ary tree of depth d. This tree will have n^d leaves, 1 root and $K_n^d = \frac{n^d - n}{n-1} \sim n^{d-1}$ internal nodes. We forget any modeling issues we might have, but focus on the symmetricity assumption in `Distmult`.

Leaves and the root only appear on one side of the queries (i, p, j) and hence won't have any problems with the symmetricity. We now focus on an internal node i. It has n children $(c_k^i)_{k=1..n}$ and one ancestor a_i. Assuming $n > 2$, the MRR associated with this node will be higher if the query $(i, p, ?)$ yields the ranked list $[c_1^i, ..., c_n^i, a_i]$. Indeed, the filtered rank of the n queries (i, p, c_k^i) will be 1 while the filtered rank of the query (a_i, p, i) will be $n + 1$.

Counting the number of queries for which the filtered rank is 1, we see that they far outweigh the queries for which the filtered rank is $n + 1$ in the final filtered MRR. For each internal nodes, n queries lead to a rank of 1, and only 1 to a rank of $n + 1$. For the root, n queries with a rank of 1, for the leaves, n^d queries with a rank of 1.

Our final filtered MRR is :

$$\begin{aligned} mrr &= \frac{n^d + n + K_n^d \frac{1}{n+1} + K_n^d n}{n^d + n + (n+1)K_n^d} \\ &= 1 - \underbrace{K_n^d \frac{n/(n+1)}{n^d + n + (n+1)K_n^d}}_{\sim \frac{1}{2n}} \to 1 \end{aligned}$$

Hence for big hierarchies such as hyponym or hypernym in `WN`, we expect the filtered MRR of `DistMult` to be high even though its modeling assumptions are incorrect.

8.2. Proofs

Lemma 2.

$$\min_{u \in \mathcal{U}_R(\boldsymbol{X})} \frac{1}{3} \sum_{r=1}^{R} \sum_{d=1}^{3} \|u_r^{(d)}\|_p^\alpha = \min_{u \in \mathcal{U}_R(\boldsymbol{X})} \sum_{r=1}^{R} \sqrt[3]{\prod_{d=1}^{3} \|u_r^{(d)}\|_p^\alpha}.$$

Moreover, the minimizers of the left-hand side satisfy:

$$\|u_r^{(d)}\|_p = \sqrt[3]{\prod_{d'=1}^{3} \|u_r^{(d')}\|_p}.$$

Proof. First, we characterize the minima :

$$\begin{aligned} &\min \left\{ \Omega_p^\alpha(u) \;\middle|\; u \in \mathcal{U}_R(\boldsymbol{X}) \right\} \\ &= \min \left\{ \Omega_p^\alpha(\tilde{u}) \;\middle|\; \tilde{u}_r^d = c_r^d u_r^d, u \in \mathcal{U}_R(\boldsymbol{X}), \prod_{d=1}^{3} c_r^d = 1 \right\} \\ &= \min \left\{ \frac{1}{3} \sum_{r=1}^{R} \sum_{d=1}^{3} (|c_r^d| \|u_r^{(d)}\|_p)^\alpha \;\middle|\; \right. \\ &\qquad \left. \tilde{u}_r^d = c_r^d u_r^d, u \in \mathcal{U}_R(\boldsymbol{X}), \prod_{d=1}^{3} c_r^d = 1 \right\} \end{aligned}$$

We study a summand, for $c_i, a_i > 0$:

$$\min_{\prod_{d=1}^{3} c^d = 1} \frac{1}{3} \sum_{d=1}^{3} (c_i a_i)^\alpha$$

Using constrained optimization techniques, we obtain that this minimum is obtained for :

$$c_i = \frac{\sqrt[3]{\prod_{d=1}^{3} a_i}}{a_i}$$

and has value $(\prod_{d=1}^{3} a_i)^{\alpha/3}$, which completes the proof. □

Proposition 2. *The function over third order-tensors of* $\mathbb{R}^{N_1 \times N_2 \times N_3}$ *defined as*

$$|||\boldsymbol{X}||| = \min \left\{ \|\sigma\|_{2/3} \;\middle|\; (\sigma, \tilde{u}) \in \mathcal{U}_R(\boldsymbol{X}), R \in \mathbb{N} \right\}$$

is not convex.

Proof. We first study elements of $\mathbb{R}^{2 \times 2 \times 1}$, tensors of order 3 associated with matrices of size 2×2. We have that

$$\left|\left|\left| \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \right|\right|\right| = \left|\left|\left| \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \right|\right|\right| = 1$$

Let $A = \frac{1}{2} I_2$, the mean of these two matrices. Identifying A with a $2 \times 2 \times 1$ tensor $\boldsymbol{A}$ to obtain the decomposition (σ, u) yielding $|||\boldsymbol{A}|||$, we have that the matrix A can be written as $\sum_{r=1}^{R} \sigma_r u_r^{(1)} \otimes u_r^{(2)}$. This comes from the fact that $u_r^{(3)}$ is a normalized 1×1 vector, so its only entry is equal to 1. We then write that trace $\mathrm{Tr}(A) = \sum_{r=1}^{R} \sigma_r \mathrm{Tr}(u_r^{(1)} \otimes u_r^{(2)}) \leq \sum_{r=1}^{R} \sigma_r$ by Cauchy-Schwarz. Hence $\|\sigma\|_1 \geq \mathrm{Tr}(A) = 1$. Moreover, we have $\|\sigma\|_{2/3} \geq \|\sigma\|_1$ with equality only for σ with at most one non-zero coordinate. Since A is of rank 2, its representation has at least 2 non-zero coordinates, hence $|||A||| = \|\sigma\|_{2/3} > 1$, which contradicts convexity. This proof can naturally be extended to tensors of any sizes. □

8.3. Best Published results

We report in Table 4 the references for each of the results in Table 2 in the article.

Model	Metric	Result	Reference
WN18	MRR	0.94	Trouillon et al. (2016)
	H@10	**0.97**	Ma et al. (2017)
WN18RR	MRR	0.46	Dettmers et al. (2017)
	H@10	**0.51**	Dettmers et al. (2017)
FB15K	MRR	0.84	Kadlec et al. (2017)
	H@10	**0.93**	Shen et al. (2016)
FB15K-237	MRR	0.32	Dettmers et al. (2017)
	H@10	0.49	Dettmers et al. (2017)
YAGO3-10	MRR	0.52	Dettmers et al. (2017)
	H@10	0.66	Dettmers et al. (2017)

Table 4. References for the *Best Published* row in Table 2