Complex Embeddings for Simple Link Prediction

Théo Trouillon[1,2] THEO.TROUILLON@XRCE.XEROX.COM
Johannes Welbl[3] J.WELBL@CS.UCL.AC.UK
Sebastian Riedel[3] S.RIEDEL@CS.UCL.AC.UK
Éric Gaussier[2] ERIC.GAUSSIER@IMAG.FR
Guillaume Bouchard[3] G.BOUCHARD@CS.UCL.AC.UK

[1] Xerox Research Centre Europe, 6 chemin de Maupertuis, 38240 Meylan, FRANCE
[2] Université Grenoble Alpes, 621 avenue Centrale, 38400 Saint Martin d'Hères, FRANCE
[3] University College London, Gower St, London WC1E 6BT, UNITED KINGDOM

Abstract

In statistical relational learning, the link prediction problem is key to automatically understand the structure of large knowledge bases. As in previous studies, we propose to solve this problem through latent factorization. However, here we make use of complex valued embeddings. The composition of complex embeddings can handle a large variety of binary relations, among them symmetric and antisymmetric relations. Compared to state-of-the-art models such as Neural Tensor Network and Holographic Embeddings, our approach based on *complex* embeddings is arguably *simpler*, as it only uses the Hermitian dot product, the complex counterpart of the standard dot product between real vectors. Our approach is scalable to large datasets as it remains linear in both space and time, while consistently outperforming alternative approaches on standard link prediction benchmarks.[1]

1. Introduction

Web-scale knowledge bases (KBs) provide a structured representation of world knowledge, with projects such as DBPedia (Auer et al., 2007), Freebase (Bollacker et al., 2008) or the Google Knowledge Vault (Dong et al., 2014). They enable a wide range of applications such as recommender systems, question answering or automated personal agents. The incompleteness of these KBs has stimulated research into predicting missing entries, a task known as link prediction that is one of the main problems in Statistical Relational Learning (SRL, Getoor & Taskar, 2007).

KBs express data as a directed graph with labeled edges (relations) between nodes (entities). Natural redundancies among the recorded relations often make it possible to fill in the missing entries of a KB. As an example, the relation `CountryOfBirth` is not recorded for all entities, but it can easily be inferred if the relation `CityOfBirth` is known. The goal of link prediction is the automatic discovery of such regularities. However, many relations are non-deterministic: the combination of the two facts `IsBornIn(John,Athens)` and `IsLocatedIn(Athens,Greece)` does not always imply the fact `HasNationality(John,Greece)`. Hence, it is required to handle other facts involving these relations or entities in a probabilistic fashion.

To do so, an increasingly popular method is to state the link prediction task as a 3D binary tensor completion problem, where each slice is the adjacency matrix of one relation type in the knowledge graph. Completion based on low-rank factorization or *embeddings* has been popularized with the Netflix challenge (Koren et al., 2009). A partially observed matrix or tensor is decomposed into a product of embedding matrices with much smaller rank, resulting in fixed-dimensional vector representations for each entity and relation in the database. For a given fact *r(s,o)* in which subject s is linked to object o through relation r, the score can then be recovered as a multi-linear product between the embedding vectors of s, r and o (Nickel et al., 2016a).

Binary relations in KBs exhibit various types of patterns: hierarchies and compositions like `FatherOf`, `OlderThan` or `IsPartOf`—with partial/total, strict/non-strict orders—and equivalence relations like `IsSimilarTo`. As described in Bordes et al. (2013a), a relational model should (a) be able to learn

[1] Code is available at: https://github.com/ttrouill/complex

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

all combinations of these properties, namely reflexivity/irreflexivity, symmetry/antisymmetry and transitivity, and (b) be linear in both time and memory in order to scale to the size of present day KBs, and keep up with their growth.

Dot products of embeddings scale well and can naturally handle both symmetry and (ir-)reflexivity of relations; using an appropriate loss function even enables transitivity (Bouchard et al., 2015). However, dealing with antisymmetric relations has so far almost always implied an explosion of the number of parameters (Nickel et al., 2011; Socher et al., 2013) (see Table 1), making models prone to overfitting. Finding the best ratio between expressiveness and parameter space size is the keystone of embedding models.

In this work we argue that the standard dot product between embeddings can be a very effective composition function, provided that one uses the right *representation*. Instead of using embeddings containing real numbers we discuss and demonstrate the capabilities of complex embeddings. When using complex vectors, i.e. vectors with entries in $\mathbb{C}$, the dot product is often called the *Hermitian* (or sesquilinear) dot product, as it involves the conjugate-transpose of one of the two vectors. As a consequence, the dot product is not symmetric any more, and facts about antisymmetric relations can receive different scores depending on the ordering of the entities involved. Thus complex vectors can effectively capture antisymmetric relations while retaining the efficiency benefits of the dot product, that is linearity in both space and time complexity.

The remainder of the paper is organized as follows. We first justify the intuition of using complex embeddings in the square matrix case in which there is only a single relation between entities. The formulation is then extended to a stacked set of square matrices in a third-order tensor to represent multiple relations. We then describe experiments on large scale public benchmark KBs in which we empirically show that this representation leads not only to simpler and faster algorithms, but also gives a systematic accuracy improvement over current state-of-the-art alternatives.

To give a clear comparison with respect to existing approaches using only real numbers, we also present an equivalent reformulation of our model that involves only real embeddings. This should help practitioners when implementing our method, without requiring the use of complex numbers in their software implementation.

2. Relations as Real Part of Low-Rank Normal Matrices

In this section we discuss the use of complex embeddings for low-rank matrix factorization and illustrate this by considering a simplified link prediction task with merely a single relation type.

Understanding the factorization in complex space leads to a better theoretical understanding of the class of matrices that can actually be approximated by dot products of embeddings. These are the so-called *normal matrices* for which the left and right embeddings share the same unitary basis.

2.1. Modelling Relations

Let $\mathcal{E}$ be a set of entities with $|\mathcal{E}| = n$. A relation between two entities is represented as a binary value $Y_{so} \in \{-1, 1\}$, where $s \in \mathcal{E}$ is the subject of the relation and $o \in \mathcal{E}$ its object. Its probability is given by the logistic inverse link function:

$$P(Y_{so} = 1) = \sigma(X_{so}) \tag{1}$$

where $X \in \mathbb{R}^{n \times n}$ is a latent matrix of scores, and Y the partially observed sign matrix.

Our goal is to find a generic structure for X that leads to a flexible approximation of common relations in real world KBs. Standard matrix factorization approximates X by a matrix product UV^T, where U and V are two functionally independent $n \times K$ matrices, K being the rank of the matrix. Within this formulation it is assumed that entities appearing as subjects are different from entities appearing as objects. This means that the same entity will have two different embedding vectors, depending on whether it appears as the subject or the object of a relation. This extensively studied type of model is closely related to the singular value decomposition (SVD) and fits well to the case where the matrix X is rectangular. However, in many link prediction problems, the same entity can appear as both subject *and* object. It then seems natural to learn joint embeddings of the entities, which entails sharing the embeddings of the left and right factors, as proposed by several authors to solve the link prediction problem (Nickel et al., 2011; Bordes et al., 2013b; Yang et al., 2015).

In order to use the same embedding for subjects and objects, researchers have generalised the notion of dot products to *scoring functions*, also known as *composition functions*, that combine embeddings in specific ways. We briefly recall several examples of scoring functions in Table 1, as well as the extension proposed in this paper.

Using the same embeddings for right and left factors boils down to Eigenvalue decomposition:

$$X = EWE^{-1} \ . \tag{2}$$

It is often used to approximate real symmetric matrices such as covariance matrices, kernel functions and distance or similarity matrices. In these cases all eigenvalues and eigenvectors live in the real space and E is orthogonal:

Model	Scoring Function	Relation parameters	$\mathcal{O}_{time}$	$\mathcal{O}_{space}$
RESCAL (Nickel et al., 2011)	$e_s^T W_r e_o$	$W_r \in \mathbb{R}^{K^2}$	$\mathcal{O}(K^2)$	$\mathcal{O}(K^2)$
TransE (Bordes et al., 2013b)	$\|\|(e_s + w_r) - e_o\|\|_p$	$w_r \in \mathbb{R}^K$	$\mathcal{O}(K)$	$\mathcal{O}(K)$
NTN (Socher et al., 2013)	$u_r^T f(e_s W_r^{[1..D]} e_o + V_r \begin{bmatrix} e_s \\ e_o \end{bmatrix} + b_r)$	$W_r \in \mathbb{R}^{K^2 D}, b_r \in \mathbb{R}^K$ $V_r \in \mathbb{R}^{2KD}, u_r \in \mathbb{R}^K$	$\mathcal{O}(K^2 D)$	$\mathcal{O}(K^2 D)$
DistMult (Yang et al., 2015)	$< w_r, e_s, e_o >$	$w_r \in \mathbb{R}^K$	$\mathcal{O}(K)$	$\mathcal{O}(K)$
HolE (Nickel et al., 2016b)	$w_r^T(\mathcal{F}^{-1}[\overline{\mathcal{F}[e_s]} \odot \mathcal{F}[e_o]]))$	$w_r \in \mathbb{R}^K$	$\mathcal{O}(K \log K)$	$\mathcal{O}(K)$
ComplEx	$\mathrm{Re}(< w_r, e_s, \bar{e}_o >)$	$w_r \in \mathbb{C}^K$	$\mathcal{O}(K)$	$\mathcal{O}(K)$

Table 1. Scoring functions of state-of-the-art latent factor models for a given fact $r(s, o)$, along with their relation parameters, time and space (memory) complexity. The embeddings e_s and e_o of subject s and object o are in $\mathbb{R}^K$ for each model, except for our model (ComplEx) where $e_s, e_o \in \mathbb{C}^K$. D is an additional latent dimension of the NTN model. $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote respectively the Fourier transform and its inverse, and $\odot$ is the element-wise product between two vectors.

$E^T = E^{-1}$. We are in this work however explicitly interested in problems where matrices — and thus the relations they represent — can also be antisymmetric. In that case eigenvalue decomposition is not possible in the real space; there only exists a decomposition in the complex space where embeddings $x \in \mathbb{C}^K$ are composed of a real vector component $\mathrm{Re}(x)$ and an imaginary vector component $\mathrm{Im}(x)$. With complex numbers, the dot product, also called the *Hermitian* product, or *sesquilinear* form, is defined as:

$$\langle u, v \rangle := \bar{u}^T v \tag{3}$$

where u and v are complex-valued vectors, i.e. $u = \mathrm{Re}(u) + i\mathrm{Im}(u)$ with $\mathrm{Re}(u) \in \mathbb{R}^K$ and $\mathrm{Im}(u) \in \mathbb{R}^K$ corresponding to the real and imaginary parts of the vector $u \in \mathbb{C}^K$, and i denoting the square root of -1. We see here that one crucial operation is to take the conjugate of the first vector: $\bar{u} = \mathrm{Re}(u) - i\mathrm{Im}(u)$. A simple way to justify the Hermitian product for composing complex vectors is that it provides a valid topological norm in the induced vectorial space. For example, $\bar{x}^T x = 0$ implies $x = 0$ while this is not the case for the bilinear form $x^T x$ as there are many complex vectors for which $x^T x = 0$.

Even with complex eigenvectors $E \in \mathbb{C}^{n \times n}$, the inversion of E in the eigendecomposition of Equation (2) leads to computational issues. Fortunately, mathematicians defined an appropriate class of matrices that prevents us from inverting the eigenvector matrix: we consider the space of *normal matrices*, i.e. the complex $n \times n$ matrices X, such that $X\bar{X}^T = \bar{X}^T X$. The spectral theorem for normal matrices states that a matrix X is normal if and only if it is unitarily diagonalizable:

$$X = E W \bar{E}^T \tag{4}$$

where $W \in \mathbb{C}^{n \times n}$ is the diagonal matrix of eigenvalues (with decreasing modulus) and $E \in \mathbb{C}^{n \times n}$ is a unitary matrix of eigenvectors, with $\bar{E}$ representing its complex conjugate.

The set of purely real normal matrices includes all symmetric and antisymmetric sign matrices (useful to model hierarchical relations such as `IsOlder`), as well as all orthogonal matrices (including permutation matrices), and many other matrices that are useful to represent binary relations, such as assignment matrices which represent bipartite graphs. However, far from all matrices expressed as $EW\bar{E}^T$ are purely real, and equation 1 requires the scores X to be purely real. So we simply keep only the real part of the decomposition:

$$X = \mathrm{Re}(E W \bar{E}^T) \ . \tag{5}$$

In fact, performing this projection on the real subspace allows the exact decomposition of *any* real square matrix X and not only normal ones, as shown by Trouillon et al. (2016).

Compared to the singular value decomposition, the eigenvalue decomposition has two key differences:

- The eigenvalues are not necessarily positive or real;
- The factorization (5) is useful as the rows of E can be used as vectorial representations of the entities corresponding to rows and columns of the relation matrix $\mathbf{X}$. Indeed, for a given entity, its subject embedding vector is the complex conjugate of its object embedding vector.

2.2. Low-Rank Decomposition

In a link prediction problem, the relation matrix is unknown and the goal is to recover it entirely from noisy observations. To enable the model to be *learnable*, i.e. to generalize to unobserved links, some regularity assumptions are needed. Since we deal with binary relations, we assume that they have low *sign-rank*. The sign-rank of a sign matrix is the smallest rank of a real matrix that has the same sign-pattern as Y:

$$\mathrm{rank}_{\pm}(Y) = \min_{A \in \mathbb{R}^{m \times n}} \{\mathrm{rank}(A) | \mathrm{sign}(A) = Y\} \ . \tag{6}$$

This is theoretically justified by the fact that the sign-rank is a natural complexity measure of sign matrices (Linial et al., 2007) and is linked to learnability (Alon et al., 2015), and empirically confirmed by the wide success of factorization models (Nickel et al., 2016a).

If the observation matrix Y is low-sign-rank, then our model can decompose it with a rank at most the double of the sign-rank of Y. That is, for any $Y \in \{-1,1\}^{n\times n}$, there always exists a matrix $X = \mathrm{Re}(EW\bar{E}^T)$ with the same sign pattern $\mathrm{sign}(X) = Y$, where the rank of $EW\bar{E}^T$ is at most twice the sign-rank of Y (Trouillon et al., 2016).

Although twice sounds bad, this is actually a good upper bound. Indeed, the sign-rank is often *much* lower than the rank of Y. For example, the rank of the $n \times n$ identity matrix I is n, but $\mathrm{rank}_{\pm}(I) = 3$ (Alon et al., 2015). By permutation of the columns $2j$ and $2j+1$, the I matrix corresponds to the relation `marriedTo`, a relation known to be hard to factorize (Nickel et al., 2014). Yet our model can express it in rank 6, for any n.

By imposing a low-rank $K \ll n$ on $EW\bar{E}^T$, only the first K values of $\mathrm{diag}(W)$ are non-zero. So we can directly have $E \in \mathbb{C}^{n\times K}$ and $W \in \mathbb{C}^{K\times K}$. Individual relation scores X_{so} between entities s and o can be predicted through the following product of their embeddings $e_s, e_o \in \mathbb{C}^K$:

$$X_{so} \quad = \quad \mathrm{Re}(e_s^T W \bar{e}_o) \ . \tag{7}$$

We summarize the above discussion in three points:

1. Our factorization encompasses all possible binary relations.
2. By construction, it accurately describes both symmetric and antisymmetric relations.
3. Learnable relations can be efficiently approximated by a simple low-rank factorization, using complex numbers to represent the latent factors.

3. Application to Binary Multi-Relational Data

The previous section focused on modeling a single type of relation; we now extend this model to multiple types of relations. We do so by allocating an embedding w_r to each relation r, and by sharing the entity embeddings across all relations.

Let $\mathcal{R}$ and $\mathcal{E}$ be the set of relations and entities present in the KB. We want to recover the matrices of scores $\mathbf{X}_r$ for all the relations $r \in \mathcal{R}$. Given two entities s and $o \in \mathcal{E}$, the log-odd of the probability that the fact *r(s,o)* is true is:

$$P(\mathbf{Y}_{rso} = 1) = \sigma(\phi(r, s, o; \Theta)) \tag{8}$$

where ϕ is a scoring function that is typically based on a factorization of the observed relations and Θ denotes the parameters of the corresponding model. While $\mathbf{X}$ as a whole is unknown, we assume that we observe a set of true and false facts $\{\mathbf{Y}_{rso}\}_{r(s,o)\in\Omega} \in \{-1,1\}^{|\Omega|}$, corresponding to the partially observed adjacency matrices of different relations, where $\Omega \subset \mathcal{R}\otimes\mathcal{E}\otimes\mathcal{E}$ is the set of observed triples. The goal is to find the probabilities of entries $\mathbf{Y}_{r's'o'}$ being true or false for a set of targeted unobserved triples $r'(s', o') \notin \Omega$.

Depending on the scoring function $\phi(s, r, o; \Theta)$ used to predict the entries of the tensor $\mathbf{X}$, we obtain different models. Examples of scoring functions are given in Table 1. Our model scoring function is:

$$\begin{aligned}
\phi(r, s, o; \Theta) &= \mathrm{Re}(< w_r, e_s, \bar{e}_o >) && (9)\\
&= \mathrm{Re}\Big(\sum_{k=1}^{K} w_{rk} e_{sk} \bar{e}_{ok}\Big) && (10)\\
&= \langle \mathrm{Re}(w_r), \mathrm{Re}(e_s), \mathrm{Re}(e_o)\rangle \\
&\quad + \langle \mathrm{Re}(w_r), \mathrm{Im}(e_s), \mathrm{Im}(e_o)\rangle \\
&\quad + \langle \mathrm{Im}(w_r), \mathrm{Re}(e_s), \mathrm{Im}(e_o)\rangle \\
&\quad - \langle \mathrm{Im}(w_r), \mathrm{Im}(e_s), \mathrm{Re}(e_o)\rangle && (11)
\end{aligned}$$

where $w_r \in \mathbb{C}^K$ is a complex vector . These equations provide two interesting views of the model:

- *Changing the representation*: Equation (10) would correspond to DistMult with real embeddings, but handles asymmetry thanks to the complex conjugate of one of the embeddings[2].
- *Changing the scoring function*: Equation (11) only involves real vectors corresponding to the real and imaginary parts of the embeddings and relations.

One can easily check that this function is antisymmetric when w_r is purely imaginary (i.e. its real part is zero), and symmetric when w_r is real. Interestingly, by separating the real and imaginary part of the relation embedding w_r, we obtain a decomposition of the relation matrix $\mathbf{X}_r$ as the sum of a symmetric matrix $\mathrm{Re}(E\,\mathrm{diag}(\mathrm{Re}(w_r))\bar{E}^T)$ and a antisymmetric matrix $\mathrm{Im}(E\,\mathrm{diag}(-\mathrm{Im}(w_r))\bar{E}^T)$. Relation embeddings naturally act as weights on each latent dimension: $\mathrm{Re}(w_r)$ over the symmetric, real part of $\langle e_o, e_s\rangle$, and $\mathrm{Im}(w)$ over the antisymmetric, imaginary part of $\langle e_o, e_s\rangle$. Indeed, one has $\langle e_o, e_s\rangle = \overline{\langle e_s, e_o\rangle}$, meaning that $\mathrm{Re}(\langle e_o, e_s\rangle)$ is symmetric, while $\mathrm{Im}(\langle e_o, e_s\rangle)$ is antisymmetric. This enables us to accurately describe both

[2]Note that in Equation (10) we used the standard componentwise multi-linear dot product $< a, b, c >:= \sum_k a_k b_k c_k$. This is not the Hermitian extension as it is not properly defined in the linear algebra literature.

symmetric and antisymmetric relations between pairs of entities, while still using joint representations of entities, whether they appear as subject or object of relations.

Geometrically, each relation embedding w_r is an anisotropic scaling of the basis defined by the entity embeddings E, followed by a projection onto the real subspace.

4. Experiments

In order to evaluate our proposal, we conducted experiments on both synthetic and real datasets. The synthetic dataset is based on relations that are either symmetric or antisymmetric, whereas the real datasets comprise different types of relations found in different, standard KBs. We refer to our model as ComplEx, for Complex Embeddings.

4.1. Synthetic Task

To assess the ability of our proposal to accurately model symmetry and antisymmetry, we randomly generated a KB of two relations and 30 entities. One relation is entirely symmetric, while the other is completely antisymmetric. This dataset corresponds to a $2 \times 30 \times 30$ tensor. Figure 2 shows a part of this randomly generated tensor, with a symmetric slice and an antisymmetric slice, decomposed into training, validation and test sets. The diagonal is unobserved as it is not relevant in this experiment.

The train set contains 1392 observed triples, whereas the validation and test sets contain 174 triples each. Figure 1 shows the best cross-validated Average Precision (area under Precision-Recall curve) for different factorization models of ranks ranging up to 50. Models were trained using Stochastic Gradient Descent with mini-batches and AdaGrad for tuning the learning rate (Duchi et al., 2011), by minimizing the negative log-likelihood of the logistic model with L^2 regularization on the parameters Θ of the considered model:

$$\min_{\Theta} \sum_{r(s,o)\in\Omega} \log(1+\exp(-\mathbf{Y}_{rso}\phi(s,r,o;\Theta))) + \lambda||\Theta||_2^2 \quad . \tag{12}$$

In our model, Θ corresponds to the embeddings $e_s, w_r, e_o \in \mathbb{C}^K$. We describe the full algorithm in Appendix A.

λ is validated in $\{0.1, 0.03,\ 0.01, 0.003,\ 0.001, 0.0003,\ 0.00001, 0.0\}$. As expected, DistMult (Yang et al., 2015) is not able to model antisymmetry and only predicts the symmetric relations correctly. Although TransE (Bordes et al., 2013b) is not a symmetric model, it performs poorly in practice, particularly on the antisymmetric relation. RESCAL (Nickel et al., 2011), with its large number of parameters, quickly overfits as the rank grows. Canonical Polyadic (CP) decomposition (Hitchcock, 1927) fails on both relations as it has to push symmetric and antisymmetric patterns through the entity embeddings. Surprisingly, only our model succeeds on such simple data.

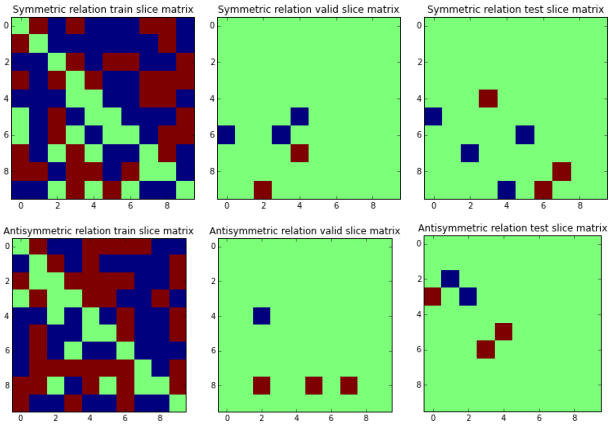

Figure 2. Parts of the training, validation and test sets of the generated experiment with one symmetric and one antisymmetric relation. Red pixels are positive triples, blue are negatives, and green missing ones. Top: Plots of the symmetric slice (relation) for the 10 first entities. Bottom: Plots of the antisymmetric slice for the 10 first entities.

4.2. Datasets: FB15K and WN18

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#triples in Train/Valid/Test
WN18	40,943	18	141,442 / 5,000 / 5,000
FB15K	14,951	1,345	483,142 / 50,000 / 59,071

Table 3. Number of entities, relations, and observed triples in each split for the FB15K and WN18 datasets.

We next evaluate the performance of our model on the FB15K and WN18 datasets. FB15K is a subset of *Freebase*, a curated KB of general facts, whereas WN18 is a subset of *Wordnet*, a database featuring lexical relations between words. We use original training, validation and test set splits as provided by Bordes et al. (2013b). Table 3 summarizes the metadata of the two datasets.

Both datasets contain only positive triples. As in Bordes et al. (2013b), we generated negatives using the *local closed world assumption*. That is, for a triple, we randomly change either the subject or the object at random, to form a negative example. This negative sampling is performed at runtime for each batch of training positive examples.

For evaluation, we measure the quality of the ranking of each test triple among all possible subject and object substitutions : $r(s', o)$ and $r(s, o')$, $\forall s', \forall o' \in \mathcal{E}$. Mean Reciprocal Rank (MRR) and Hits at m are the standard evaluation measures for these datasets and come in two flavours: raw and filtered (Bordes et al., 2013b). The filtered metrics

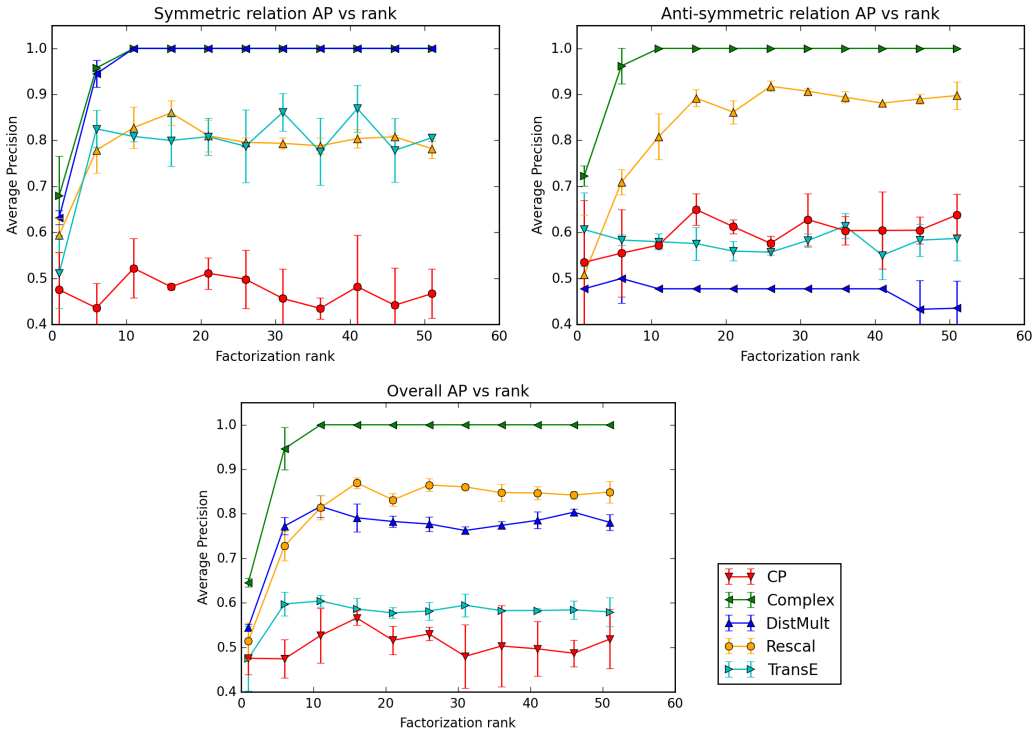

Figure 1. Average Precision (AP) for each factorization rank ranging from 1 to 50 for different state of the art models on the combined symmetry and antisymmetry experiment. Top-left: AP for the symmetric relation only. Top-right: AP for the antisymmetric relation only. Bottom: Overall AP.

are computed *after* removing all the other positive observed triples that appear in either training, validation or test set from the ranking, whereas the raw metrics do not remove these.

Since ranking measures are used, previous studies generally preferred a pairwise ranking loss for the task (Bordes et al., 2013b; Nickel et al., 2016b). We chose to use the negative log-likelihood of the logistic model, as it is a continuous surrogate of the sign-rank, and has been shown to learn compact representations for several important relations, especially for transitive relations (Bouchard et al., 2015). In preliminary work, we tried both losses, and indeed the log-likelihood yielded better results than the ranking loss (except with TransE), especially on FB15K.

We report both filtered and raw MRR, and filtered Hits at 1, 3 and 10 in Table 2 for the evaluated models. Furthermore, we chose TransE, DistMult and HolE as baselines since they are the best performing models on those datasets to the best of our knowledge (Nickel et al., 2016b; Yang et al., 2015). We also compare with the CP model to emphasize empirically the importance of learning unique embeddings for entities. For experimental fairness, we reimplemented these methods within the same framework as the ComplEx model, using theano (Bergstra et al., 2010). However, due to time constraints and the complexity of an efficient implementation of HolE, we record the original results for HolE as reported in Nickel et al. (2016b).

4.3. Results

WN18 describes lexical and semantic hierarchies between concepts and contains many antisymmetric relations such as hypernymy, hyponymy, or being "part of". Indeed, the DistMult and TransE models are outperformed here by ComplEx and HolE, which are on par with respective filtered MRR scores of 0.941 and 0.938. Table 4 shows the filtered test set MRR for the models considered and each relation of WN18, confirming the advantage of our model on antisymmetric relations while losing nothing on the others. 2D projections of the relation embeddings provided in Appendix B visually corroborate the results.

On FB15K, the gap is much more pronounced and the ComplEx model largely outperforms HolE, with a filtered MRR of 0.692 and 59.9% of Hits at 1, compared to 0.524 and 40.2% for HolE. We attribute this to the simplicity of our model and the different loss function. This is supported by the relatively small gap in MRR compared to DistMult (0.654); our model can in fact be interpreted as a complex number version of DistMult. On both datasets, TransE

	WN18					FB15K				
	MRR		Hits at			MRR		Hits at		
Model	Filter	Raw	1	3	10	Filter	Raw	1	3	10
CP	0.075	0.058	0.049	0.080	0.125	0.326	0.152	0.219	0.376	0.532
TransE	0.454	0.335	0.089	0.823	0.934	0.380	0.221	0.231	0.472	0.641
DistMult	0.822	0.532	0.728	0.914	0.936	0.654	**0.242**	0.546	0.733	0.824
HolE*	0.938	**0.616**	0.93	**0.945**	**0.949**	0.524	0.232	0.402	0.613	0.739
ComplEx	**0.941**	0.587	**0.936**	**0.945**	0.947	**0.692**	**0.242**	**0.599**	**0.759**	**0.840**

Table 2. Filtered and Raw Mean Reciprocal Rank (MRR) for the models tested on the FB15K and WN18 datasets. Hits@m metrics are filtered. *Results reported from (Nickel et al., 2016b) for HolE model.

Relation name	ComplEx	DistMult	TransE
hypernym	**0.953**	0.791	0.446
hyponym	**0.946**	0.710	0.361
member_meronym	**0.921**	0.704	0.418
member_holonym	**0.946**	0.740	0.465
instance_hypernym	**0.965**	0.943	0.961
instance_hyponym	**0.945**	0.940	0.745
has_part	**0.933**	0.753	0.426
part_of	**0.940**	0.867	0.455
member_of_domain_topic	**0.924**	0.914	0.861
synset_domain_topic_of	**0.930**	0.919	0.917
member_of_domain_usage	**0.917**	**0.917**	0.875
synset_domain_usage_of	**1.000**	**1.000**	**1.000**
member_of_domain_region	**0.865**	0.635	**0.865**
synset_domain_region_of	0.919	0.888	**0.986**
derivationally_related_form	**0.946**	0.940	0.384
similar_to	**1.000**	**1.000**	0.244
verb_group	**0.936**	0.897	0.323
also_see	0.603	**0.607**	0.279

Table 4. Filtered Mean Reciprocal Rank (MRR) for the models tested on each relation of the Wordnet dataset (WN18).

and CP are largely left behind. This illustrates the power of the simple dot product in the first case, and the importance of learning unique entity embeddings in the second. CP performs poorly on WN18 due to the small number of relations, which magnifies this subject/object difference.

Reported results are given for the best set of hyper-parameters evaluated on the validation set for each model, after grid search on the following values: $K \in \{10, 20, 50, 100, 150, 200\}$, $\lambda \in \{0.1, 0.03, 0.01, 0.003, 0.001, 0.0003, 0.0\}$, $\alpha_0 \in \{1.0, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01\}$, $\eta \in \{1, 2, 5, 10\}$ with λ the L^2 regularization parameter, α_0 the initial learning rate (then tuned at runtime with AdaGrad), and η the number of negatives generated per positive training triple. We also tried varying the batch size but this had no impact and we settled with 100 batches per epoch. Best ranks were generally 150 or 200, in both cases scores were always very close for all models. The number of negative samples per positive sample also had a large influence on the filtered MRR on FB15K (up to +0.08 improvement from 1 to 10 negatives), but not much on WN18. On both datasets regularization was important (up to +0.05 on filtered MRR between $\lambda = 0$ and optimal one). We found the initial learning rate to be very important on FB15K, while not so much on WN18. We think this may also explain the large gap of improvement our model provides on this dataset compared to previously published results – as DistMult results are also better than those previously reported (Yang et al., 2015) – along with the use of the log-likelihood objective. It seems that in general AdaGrad is relatively insensitive to the initial learning rate, perhaps causing some overconfidence in its ability to tune the step size online and consequently leading to less efforts when selecting the initial step size.

Training was stopped using early stopping on the validation set filtered MRR, computed every 50 epochs with a maximum of 1000 epochs.

4.4. Influence of Negative Samples

We further investigated the influence of the number of negatives generated per positive training sample. In the previous experiment, due to computational limitations, the number of negatives per training sample, η, was validated among the possible numbers $\{1, 2, 5, 10\}$. We want to explore here whether increasing these numbers could lead to better results. To do so, we focused on FB15K, with the best validated λ, K, α_0, obtained from the previous experiment. We then let η vary in $\{1, 2, 5, 10, 20, 50, 100, 200\}$.

Figure 3 shows the influence of the number of generated negatives per positive training triple on the performance of our model on FB15K. Generating more negatives clearly improves the results, with a filtered MRR of 0.737 with 100 negative triples (and 64.8% of Hits@1), before decreasing again with 200 negatives. The model also converges with fewer epochs, which compensates partially for the additional training time per epoch, up to 50 negatives. It then grows linearly as the number of negatives increases, making 50 a good trade-off between accuracy and training time.

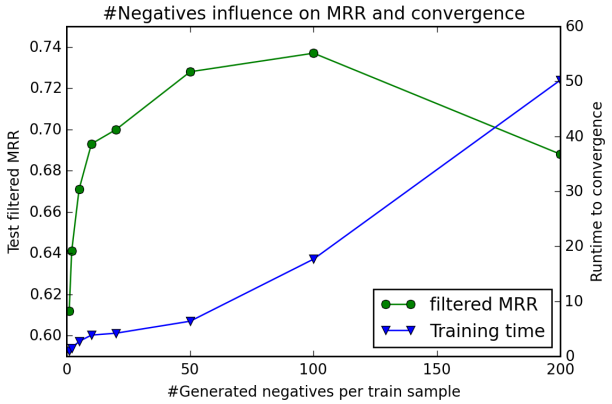

Figure 3. Influence of the number of negative triples generated per positive training example on the filtered test MRR and on training time to convergence on FB15K for the ComplEx model with $K = 200$, $\lambda = 0.01$ and $\alpha_0 = 0.5$. Times are given relative to the training time with one negative triple generated per positive training sample ($= 1$ on time scale).

5. Related Work

In the early age of spectral theory in linear algebra, complex numbers were not used for matrix factorization and mathematicians mostly focused on bi-linear forms (Beltrami, 1873). The eigen-decomposition in the complex domain as taught today in linear algebra courses came 40 years later (Autonne, 1915). Similarly, most of the existing approaches for tensor factorization were based on decompositions in the real domain, such as the Canonical Polyadic (CP) decomposition (Hitchcock, 1927). These methods are very effective in many applications that use different modes of the tensor for different types of entities. But in the link prediction problem, antisymmetry of relations was quickly seen as a problem and asymmetric extensions of tensors were studied, mostly by either considering independent embeddings (Sutskever, 2009) or considering relations as matrices instead of vectors in the RESCAL model (Nickel et al., 2011). Direct extensions were based on uni-,bi- and trigram latent factors for triple data, as well as a low-rank relation matrix (Jenatton et al., 2012).

Pairwise interaction models were also considered to improve prediction performances. For example, the Universal Schema approach (Riedel et al., 2013) factorizes a 2D unfolding of the tensor (a matrix of entity pairs vs. relations) while Welbl et al. (2016) extend this also to other pairs.

In the Neural Tensor Network (NTN) model, Socher et al. (2013) combine linear transformations and multiple bilinear forms of subject and object embeddings to jointly feed them into a nonlinear neural layer. Its non-linearity and multiple ways of including interactions between embeddings gives it an advantage in expressiveness over models with simpler scoring function like DistMult or RESCAL. As a downside, its very large number of parameters can make the NTN model harder to train and overfit more easily.

The original multi-linear DistMult model is symmetric in subject and object for every relation (Yang et al., 2015) and achieves good performance, presumably due to its simplicity. The TransE model from Bordes et al. (2013b) also embeds entities and relations in the same space and imposes a geometrical structural bias into the model: the subject entity vector should be close to the object entity vector once translated by the relation vector.

A recent novel way to handle antisymmetry is via the Holographic Embeddings (HolE) model by (Nickel et al., 2016b). In HolE the circular correlation is used for combining entity embeddings, measuring the covariance between embeddings at different dimension shifts. This generally suggests that other composition functions than the classical tensor product can be helpful as they allow for a richer interaction of embeddings. However, the asymmetry in the composition function in HolE stems from the asymmetry of circular correlation, an $\mathcal{O}(nlog(n))$ operation, whereas ours is inherited from the complex inner product, in $\mathcal{O}(n)$.

6. Conclusion

We described a simple approach to matrix and tensor factorization for link prediction data that uses vectors with complex values and retains the mathematical definition of the dot product. The class of normal matrices is a natural fit for binary relations, and using the real part allows for efficient approximation of any learnable relation. Results on standard benchmarks show that no more modifications are needed to improve over the state-of-the-art.

There are several directions in which this work can be extended. An obvious one is to merge our approach with known extensions to tensor factorization in order to further improve predictive performance. For example, the use of pairwise embeddings together with complex numbers might lead to improved results in many situations that involve non-compositionality. Another direction would be to develop a more intelligent negative sampling procedure, to generate more informative negatives with respect to the positive sample from which they have been sampled. It would reduce the number of negatives required to reach good performance, thus accelerating training time.

Also, if we were to use complex embeddings every time a model includes a dot product, e.g. in deep neural networks, would it lead to a similar systematic improvement?

Acknowledgements

This work was supported in part by the Paul Allen Foundation through an Allen Distinguished Investigator grant and in part by a Google Focused Research Award.

References

Alon, Noga, Moran, Shay, and Yehudayoff, Amir. Sign rank versus vc dimension. *arXiv preprint arXiv:1503.07648*, 2015.

Auer, Sren, Bizer, Christian, Kobilarov, Georgi, Lehmann, Jens, and Ives, Zachary. Dbpedia: A nucleus for a web of open data. In *In 6th Intl Semantic Web Conference, Busan, Korea*, pp. 11–15. Springer, 2007.

Autonne, L. Sur les matrices hypohermitiennes et sur les matrices unitaires. *Ann. Univ. Lyons, Nouvelle Srie I*, 38: 1–77, 1915.

Beltrami, Eugenio. Sulle funzioni bilineari. *Giornale di Matematiche ad Uso degli Studenti Delle Universita*, 11 (2):98–106, 1873.

Bergstra, James, Breuleux, Olivier, Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Desjardins, Guillaume, Turian, Joseph, Warde-Farley, David, and Bengio, Yoshua. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

Bollacker, Kurt, Evans, Colin, Paritosh, Praveen, Sturge, Tim, and Taylor, Jamie. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD 08 Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008.

Bordes, Antoine, Usunier, Nicolas, Garcia-Duran, Alberto, Weston, Jason, and Yakhnenko, Oksana. Irreflexive and Hierarchical Relations as Translations. In *CoRR*, 2013a.

Bordes, Antoine, Usunier, Nicolas, Garcia-Duran, Alberto, Weston, Jason, and Yakhnenko, Oksana. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pp. 2787–2795, 2013b.

Bouchard, Guillaume, Singh, Sameer, and Trouillon, Théo. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*, 2015.

Dong, Xin, Gabrilovich, Evgeniy, Heitz, Geremy, Horn, Wilko, Lao, Ni, Murphy, Kevin, Strohmann, Thomas, Sun, Shaohua, and Zhang, Wei. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 601–610, 2014.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Getoor, Lise and Taskar, Ben. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007. ISBN 0262072882.

Hitchcock, F. L. The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys*, 6(1):164–189, 1927.

Jenatton, Rodolphe, Bordes, Antoine, Le Roux, Nicolas, and Obozinski, Guillaume. A Latent Factor Model for Highly Multi-relational Data. In *Advances in Neural Information Processing Systems 25*, pp. 3167–3175, 2012.

Koren, Yehuda, Bell, Robert, and Volinsky, Chris. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Linial, Nati, Mendelson, Shahar, Schechtman, Gideon, and Shraibman, Adi. Complexity measures of sign matrices. *Combinatorica*, 27(4):439–463, 2007.

Nickel, Maximilian, Tresp, Volker, and Kriegel, Hans-Peter. A Three-Way Model for Collective Learning on Multi-Relational Data. In *28th International Conference on Machine Learning*, pp. 809—-816, 2011.

Nickel, Maximilian, Jiang, Xueyan, and Tresp, Volker. Reducing the rank in relational factorization models by including observable patterns. In *Advances in Neural Information Processing Systems*, pp. 1179–1187, 2014.

Nickel, Maximilian, Murphy, Kevin, Tresp, Volker, and Gabrilovich, Evgeniy. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016a.

Nickel, Maximilian, Rosasco, Lorenzo, and Poggio, Tomaso A. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1955–1961, 2016b.

Riedel, Sebastian, Yao, Limin, McCallum, Andrew, and Marlin, Benjamin M. Relation extraction with matrix factorization and universal schemas. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*, pp. 74–84, 2013.

Socher, Richard, Chen, Danqi, Manning, Christopher D, and Ng, Andrew. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pp. 926–934, 2013.

Sutskever, Ilya. Modelling Relational Data using Bayesian Clustered Tensor Factorization. In *Advances in Neural Information Processing Systems*, volume 22, pp. 1–8, 2009.

Trouillon, Théo, Dance, Christopher R., Gaussier, Éric, and Bouchard, Guillaume. Decomposing real square matrices via unitary diagonalization. *arXiv:1605.07103*, 2016.

Welbl, Johannes, Bouchard, Guillaume, and Riedel, Sebastian. A factorization machine framework for testing bigram embeddings in knowledgebase completion. *arXiv:1604.05878*, 2016.

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

A. SGD algorithm

We describe the algorithm to learn the ComplEx model with Stochastic Gradient Descent using only real-valued vectors.

Let us rewrite equation 11, by denoting the real part of embeddings with primes and the imaginary part with double primes: $e'_i = \mathrm{Re}(e_i)$, $e''_i = \mathrm{Im}(e_i)$, $w'_r = \mathrm{Re}(w_r)$, $w''_r = \mathrm{Im}(w_r)$. The set of parameters is $\Theta = \{e'_i, e''_i, w'_r, w''_r; \forall i \in \mathcal{E}, \forall r \in \mathcal{R}\}$, and the scoring function involves only real vectors:

$$\begin{aligned}\phi(r,s,o;\Theta) &= \langle w'_r, e'_s, e'_o\rangle + \langle w'_r, e''_s, e''_o\rangle \\ &\quad + \langle w''_r, e'_s, e''_o\rangle - \langle w''_r, e''_s, e'_o\rangle\end{aligned}$$

where each entity and each relation has two real embeddings.

Gradients are now easy to write:

$$\begin{aligned}\nabla_{e'_s}\phi(r,s,o;\Theta) &= (w'_r \odot e'_o) + (w''_r \odot e''_o)\\ \nabla_{e''_s}\phi(r,s,o;\Theta) &= (w'_r \odot e''_o) - (w''_r \odot e'_o)\\ \nabla_{e'_o}\phi(r,s,o;\Theta) &= (w'_r \odot e'_s) - (w''_r \odot e''_s)\\ \nabla_{e''_o}\phi(r,s,o;\Theta) &= (w'_r \odot e''_s) + (w''_r \odot e'_s)\\ \nabla_{w'_r}\phi(r,s,o;\Theta) &= (e'_s \odot e'_o) + (e''_s \odot e''_o)\\ \nabla_{w''_r}\phi(r,s,o;\Theta) &= (e'_s \odot e''_o) - (e''_s \odot e'_o)\end{aligned}$$

where $\odot$ is the element-wise (Hadamard) product.

As stated in equation 8 we use the sigmoid link function, and minimize the L^2-regularized negative log-likelihood:

$$\begin{aligned}\gamma(\Omega;\Theta) &= \sum_{r(s,o)\in\Omega} \log(1+\exp(-\mathbf{Y}_{rso}\phi(s,r,o;\Theta))) \\ &\quad +\lambda||\Theta||_2^2 \ .\end{aligned}$$

To handle regularization, note that the squared L^2-norm of a complex vector $v = v' + iv''$ is the sum of the squared modulus of each entry:

$$\begin{aligned}||v||_2^2 &= \sum_j \sqrt{v'^2_j + v''^2_j}^2 \\ &= \sum_j v'^2_j + \sum_j v''^2_j \\ &= ||v'||_2^2 + ||v''||_2^2\end{aligned}$$

which is actually the sum of the L^2-norms of the vectors of the real and imaginary parts.

Algorithm 1 SGD for the ComplEx model

input Training set Ω, Validation set Ω_v, learning rate α, embedding dim. k, regularization factor λ, negative ratio η, batch size b, max iter m, early stopping s.
  $e'_i \leftarrow \mathrm{randn}(k)$, $e''_i \leftarrow \mathrm{randn}(k)$ for each $i \in \mathcal{E}$
  $w'_i \leftarrow \mathrm{randn}(k)$, $w''_i \leftarrow \mathrm{randn}(k)$ for each $i \in \mathcal{R}$
  for $i = 1, \cdots, m$ **do**
    for $j = 1..|\Omega|/b$ **do**
      $\Omega_b \leftarrow \mathrm{sample}(\Omega, b, \eta)$
      Update embeddings w.r.t.:
        $\sum_{r(s,o)\in\Omega_b} \nabla\gamma(\{r(s,o)\};\Theta)$
      Update learning rate α using Adagrad
    end for
    if $i \mod s = 0$ **then**
      break if filteredMRR or AP on Ω_v decreased
    end if
  end for

We can finally write the gradient of γ with respect to a *real* embedding v for one triple $r(s,o)$:

$$\begin{aligned}\nabla_v\gamma(\{r(s,o)\};\Theta) &= -\mathbf{Y}_{rso}\phi(s,r,o;\Theta)\sigma(\nabla_v\phi(r,s,o;\Theta)) \\ &\quad +2\lambda v\end{aligned}$$

where $\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$ is the sigmoid function.

Algorithm 1 describes SGD for this formulation of the scoring function. When Ω contains only positive triples, we generate η negatives per positive train triple, by corrupting either the subject or the object of the positive triple, as described in Bordes et al. (2013b).

B. WN18 embeddings visualization

We used principal component analysis (PCA) to visualize embeddings of the relations of the wordnet dataset (WN18). We plotted the four first components of the best DistMult and ComplEx model's embeddings in Figure 4. For the ComplEx model, we simply concatenated the real and imaginary parts of each embedding.

Most of WN18 relations describe hierarchies, and are thus antisymmetric. Each of these hierarchic relations has its inverse relation in the dataset. For example: `hypernym` / `hyponym`, `part_of` / `has_part`, `synset_domain_topic_of` / `member_of_domain_topic`. Since DistMult is unable to model antisymmetry, it will correctly represent the nature of each pair of opposite relations, but not the direction of the relations. Loosely speaking, in the `hypernym` / `hyponym` pair the nature is sharing semantics, and the direction is that one entity generalizes the semantics of the other. This makes DistMult reprensenting the opposite

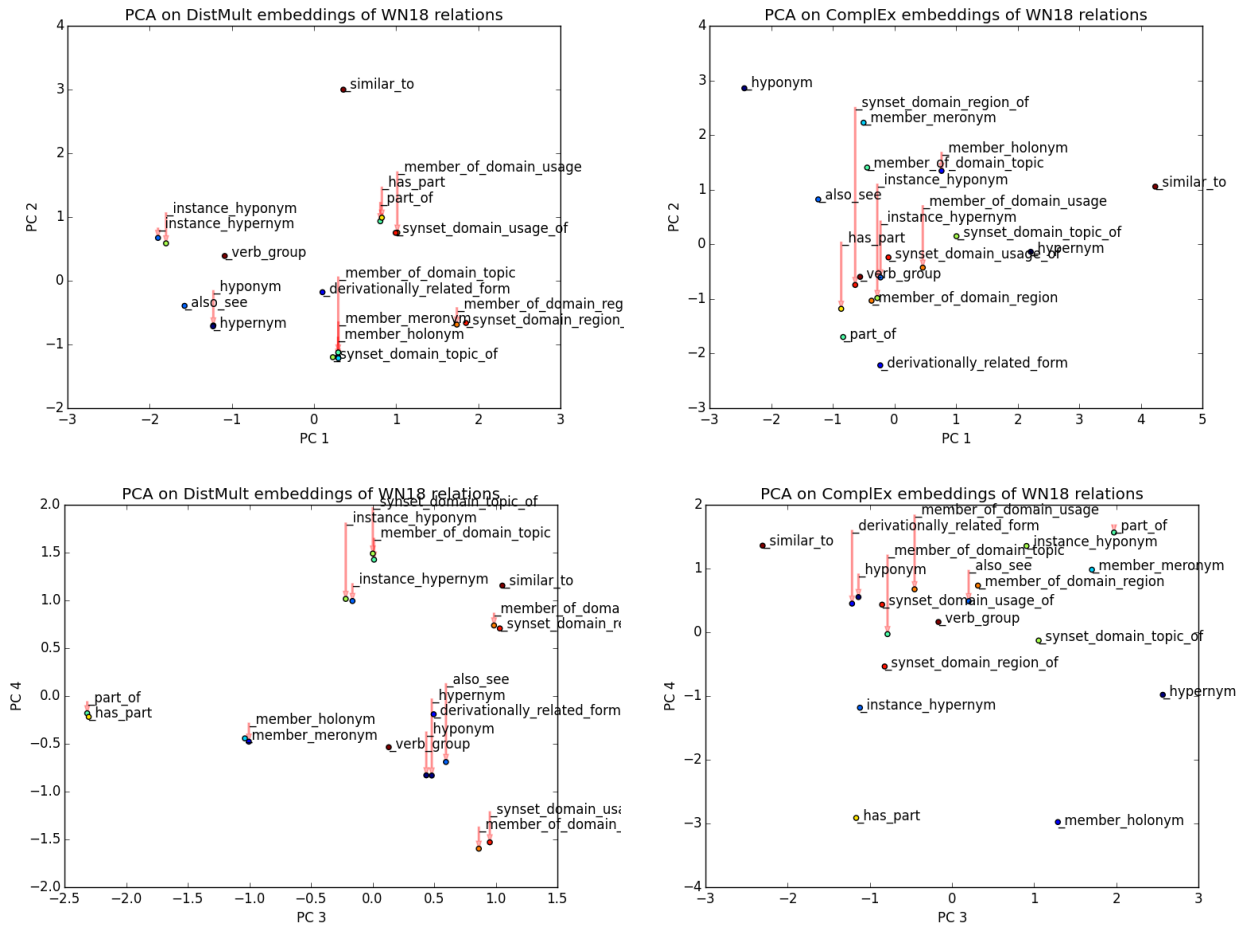

Figure 4. Plots of the first and second (Top), third and fourth (Bottom) components of the WN18 relations embeddings using PCA. Left: DistMult embeddings. Right: ComplEx embeddings. Opposite relations are clustered together by DistMult while correctly separated by ComplEx.

relations with very close embeddings, as Figure 4 shows. It is especially striking for the third and fourth principal component (bottom-left). Conversely, ComplEx manages to oppose spatially the opposite relations.