Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Holographic Embeddings of Knowledge Graphs

Maximilian Nickel[1,2] and **Lorenzo Rosasco**[1,2,3] and **Tomaso Poggio**[1]
[1]Laboratory for Computational and Statistical Learning and Center for Brains, Minds and Machines
Massachusetts Institute of Technology, Cambridge, MA
[2]Istituto Italiano di Tecnologia, Genova, Italy
[3]DIBRIS, Universita Degli Studi Di Genova, Italy

Abstract

Learning embeddings of entities and relations is an efficient and versatile method to perform machine learning on relational data such as knowledge graphs. In this work, we propose holographic embeddings (HOLE) to learn compositional vector space representations of entire knowledge graphs. The proposed method is related to holographic models of associative memory in that it employs circular correlation to create compositional representations. By using correlation as the compositional operator, HOLE can capture rich interactions but simultaneously remains efficient to compute, easy to train, and scalable to very large datasets. Experimentally, we show that holographic embeddings are able to outperform state-of-the-art methods for link prediction on knowledge graphs and relational learning benchmark datasets.

Introduction

Relations are a key concept in artificial intelligence and cognitive science. Many of the structures that humans impose on the world, such as logical reasoning, analogies, or taxonomies, are based on entities, concepts and their relationships. Hence, learning from and with relational knowledge representations has long been considered an important task in artificial intelligence (see e.g., (Getoor and Taskar 2007; Muggleton 1991; Gentner 1983; Kemp et al. 2006; Xu et al. 2006; Richardson and Domingos 2006)). In this work we are concerned with learning from knowledge graphs (KGs), i.e., knowledge bases which model facts as instances of binary relations (e.g., *bornIn(BarackObama, Hawaii)*). This form of knowledge representation can be interpreted as a multigraph, where entities correspond to nodes, facts correspond to typed edges, and the type of an edge indicates the kind of the relation. Modern knowledge graphs such as YAGO (Suchanek, Kasneci, and Weikum 2007), DBpedia (Auer et al. 2007), and Freebase (Bollacker et al. 2008) contain billions of facts about millions of entities and have found important applications in question answering, structured search, and digital assistants. Recently, vector space embeddings of knowledge graphs have received considerable attention, as they can be used to create statistical models of entire KGs, i.e., to predict the probability of any possible relation instance (edge) in the graph. Such models can be used to derive new knowledge from known facts (link prediction), to disambiguate entities (entity resolution), to extract taxonomies, and for probabilistic question answering (see e.g., (Nickel, Tresp, and Kriegel 2011; Bordes et al. 2013; Krompaß, Nickel, and Tresp 2014)). Furthermore, embeddings of KGs have been used to support machine reading and to assess the trustworthiness of web sites (Dong et al. 2014; 2015). However, existing embedding models that can capture rich interactions in relational data are often limited in their scalability. Vice versa, models that can be computed efficiently are often considerably less expressive. In this work, we approach learning from KGs within the framework of compositional vector space models. We introduce holographic embeddings (HOLE) which use the circular correlation of entity embeddings (vector representations) to create compositional representations of binary relational data. By using correlation as the compositional operator HOLE can capture rich interactions but simultaneously remains efficient to compute, easy to train, and scalable to very large datasets. As we will show experimentally, HOLE is able to outperform state-of-the-art embedding models on various benchmark datasets for learning from KGs. Compositional vector space models have also been considered in cognitive science and natural language processing, e.g., to model symbolic structures, to represent the semantic meaning of phrases, and as models for associative memory (see e.g., (Smolensky 1990; Plate 1995; Mitchell and Lapata 2008; Socher et al. 2012)). In this work, we do not only draw inspiration from these models, but we will also highlight the connections of HOLE to holographic models of associative memory.

Compositional Representations

In this section we introduce compositional vector space models for KGs, the general learning setting, and related work.

Let $\mathcal{E}$ denote the set of all entities and $\mathcal{P}$ the set of all relation types (predicates) in a domain. A binary relation is a subset $\mathcal{R}_p \subseteq \mathcal{E} \times \mathcal{E}$ of all pairs of entities (i.e., those pairs which are in a relation of type p). Higher-arity relations are defined analogously. The characteristic function $\phi_p : \mathcal{E} \times \mathcal{E} \to \{\pm 1\}$ of a relation $\mathcal{R}_p$ indicates for each *possible* pair of entities whether they are part of $\mathcal{R}_p$. We will denote (possible) relation instances as $\mathcal{R}_p(\mathsf{s}, \mathsf{o})$, where $\mathsf{s}, \mathsf{o} \in \mathcal{E}$ denote the first and second argument of the asymmetric relation $\mathcal{R}_p$. We will refer to s, o as *subject* and *object* and to $\mathcal{R}_p(\mathsf{s}, \mathsf{o})$ as *triples*.

Compositional vector space models provide an elegant way to learn the characteristic functions of the relations in a knowledge graph, as they allow to cast the learning task as a problem of supervised representation learning. Here, we discuss models of the form

$$\Pr(\phi_p(\mathsf{s}, \mathsf{o}) = 1 | \Theta) = \sigma(\eta_{spo}) = \sigma(\boldsymbol{r}_p^\top(\boldsymbol{e}_s \circ \boldsymbol{e}_o)) \tag{1}$$

where $\boldsymbol{r}_p \in \mathbb{R}^{d_r}$, $\boldsymbol{e}_i \in \mathbb{R}^{d_e}$ are vector representations of relations and entities; $\sigma(x) = 1/(1 + \exp(-x))$ denotes the logistic function; $\Theta = \{\boldsymbol{e}_i\}_{i=1}^{n_e} \cup \{\boldsymbol{r}_k\}_{k=1}^{n_r}$ denotes the set of all embeddings; $\circ : \mathbb{R}^{d_e} \times \mathbb{R}^{d_e} \to \mathbb{R}^{d_p}$ denotes the compositional operator which creates a *composite* vector representation for the pair (s, o) from the embeddings $\boldsymbol{e}_s, \boldsymbol{e}_o$. We will discuss possible compositional operators below.

Let $x_i \in \mathcal{P} \times \mathcal{E} \times \mathcal{E}$ denote a triple, and $y_i \in \{\pm 1\}$ denote its label. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ of true and false relation instances, we then want to learn representations of entities and relations Θ that best explain $\mathcal{D}$ according to eq. (1). This can, for instance, be done by minimizing the (regularized) logistic loss

$$\min_{\Theta} \sum_{i=1}^{m} \log(1 + \exp(-y_i \eta_i)) + \lambda \|\Theta\|_2^2. \tag{2}$$

For relational data, minimizing the logistic loss has the additional advantage that it can help to find low dimensional embeddings for complex relational patterns (Bouchard, Singh, and Trouillon 2015). However, in most cases, KGs store only true triples and non-existing triples can be either missing of false (open-world assumption). In this case, negative examples can be generated by heuristics such as the local closed world assumption (Dong et al. 2014). Alternatively, we can use a pairwise ranking loss such as

$$\min_{\Theta} \sum_{i \in \mathcal{D}_+} \sum_{j \in \mathcal{D}_-} \max(0, \gamma + \sigma(\eta_j) - \sigma(\eta_i)) \tag{3}$$

to rank the probability of existing triples higher than the probability of non-existing ones. Here, $\mathcal{D}^+, \mathcal{D}^-$ denote the set of existing and non-existing triples and $\gamma > 0$ specifies the width of the margin (Bordes et al. 2011).

An important property of compositional models is that the meaning and representation of entities does *not* vary with regard to their position in the compositional representation (i.e., the i-th entity has the same representation $\boldsymbol{e}_i$ as subject and object). Since the representations of all entities and relations are learned jointly in eqs. (2) and (3), this property allows to propagate information between triples, to capture global dependencies in the data, and to enable the desired relational learning effect. For a review of machine learning on knowledge graphs see also Nickel et al. (2015).

Existing models for knowledge graphs are based on the following compositional operators:

Tensor Product Given entity embeddings $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, tensor product models represent pairs of entities via $\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{a} \otimes \boldsymbol{b} \in \mathbb{R}^{d^2}$, i.e, via all pairwise multiplicative interactions between the features of $\boldsymbol{a}$ and $\boldsymbol{b}$:

$$[\boldsymbol{a} \otimes \boldsymbol{b}]_{ij} = a_i b_j. \tag{4}$$

Intuitively, a feature in the tuple representation $\boldsymbol{a} \otimes \boldsymbol{b}$ is “on” (has a high absolute magnitude), if and only if the corresponding features of *both* entities are “on” (See also fig. 1a). This allows eq. (1) to capture relational patterns such as *liberal persons are typically members of liberal parties* since a single feature in $\boldsymbol{a} \otimes \boldsymbol{b}$ can encode that the subject is a liberal person and that the object is a liberal party. Compositional models using the tensor product such as RESCAL (Nickel, Tresp, and Kriegel 2011) and the Neural Tensor Network (Socher et al. 2013). have shown state-of-the-art performance for learning from KGs. Furthermore, Guu, Miller, and Liang (2015) proposed a RESCAL-based model to learn from paths in KGs. Smolensky (1990) introduced the tensor product as a way to create compositional vector representations. While the tensor product allows to capture rich interactions, its main problem as a compositional operator lies in the fact that it requires a large number of parameters. Since $\boldsymbol{a} \otimes \boldsymbol{b}$ explicitly models all pairwise interactions, $\boldsymbol{r}_p$ in eq. (1) must be of size d^2. This can be problematic both in terms of overfitting and computational demands. For instance, Nickel, Jiang, and Tresp (2014) showed that linear tensor factorization can require large d to model certain relations. Since $\boldsymbol{r}_p^\top(\boldsymbol{e}_s \otimes \boldsymbol{e}_o) = \boldsymbol{e}_s^\top R_p \boldsymbol{e}_o$, Yang et al. (2015) proposed to use diagonal R_p's to reduce the number of parameters. However, this approach can only model symmetric relations and is not suitable to model general knowledge graphs as $\boldsymbol{e}_s^\top R_p \boldsymbol{e}_o = \boldsymbol{e}_o^\top R_p \boldsymbol{e}_s$ for diagonal R_p.

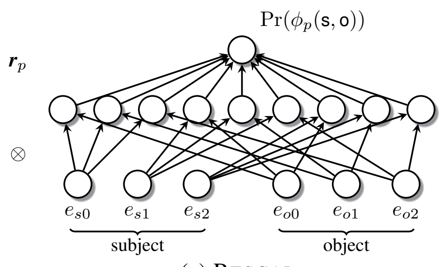
Concatenation, Projection, and Non-Linearity Another way to compute composite representations is via concatenation, projection and subsequent application of a non-linear function. Let $\oplus : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}^{d_1+d_2}$ denote concatenation and $\psi : \mathbb{R} \to \mathbb{R}$ be a non-linear function such as *tanh*. The composite tuple representation is then given by $\boldsymbol{a} \circ \boldsymbol{b} = \psi(W(\boldsymbol{a} \oplus \boldsymbol{b})) \in \mathbb{R}^h$, such that

$$[\psi(W(\boldsymbol{a} \oplus \boldsymbol{b}))]_i = \psi\left(\sum\nolimits_j w_{ij}^a a_j + \sum\nolimits_j w_{ij}^b b_j\right) \tag{5}$$

where the projection matrix $W \in \mathbb{R}^{h \times 2d}$ is learned in combination with the entity and relation embeddings. Intuitively, a feature in the tuple representation $W(\boldsymbol{a} \oplus \boldsymbol{b})$ is “on” if *at least one* of the corresponding features is “on”. An advantage of this compositional operator is that the mapping from entity embeddings to representations of pairs is learned adaptively via the matrix W. However, the resulting composite representations are also less rich, as they do not consider direct interactions of features. As Socher et al. (2013) noted, the non-linearity ψ provides only weak interactions while leading to a harder optimization problem. A variant of this compositional operator which also includes a relation embedding in the composite representation has been used in the ER-MLP model of the Knowledge Vault (Dong et al. 2014).

Non-compositional Methods Another class of models does not (explicitly) form compositional representations, but predicts the existence of triples from the similarity of the vector space embeddings. In particular, TRANSE (Bordes et al. 2013) models the score of a fact as the distance between relation-specific translations of entity embeddings:

$$\operatorname{score}(\mathcal{R}_p(\mathsf{s}, \mathsf{o})) = -\operatorname{dist}(\boldsymbol{e}_{\mathrm{s}} + \boldsymbol{r}_{\mathrm{p}}, \boldsymbol{e}_{\mathrm{o}}) \,. \tag{6}$$


$\Pr(\phi_p(\mathsf{s},\mathsf{o}))$
$\boldsymbol{r}_p$
$\otimes$
e_{s0} e_{s1} e_{s2} e_{o0} e_{o1} e_{o2}
subject
object
(a) RESCAL

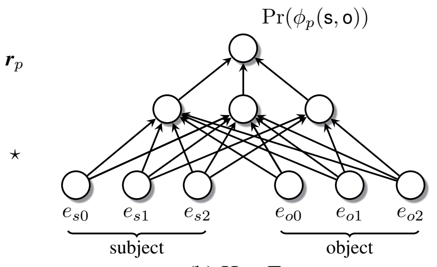

$\Pr(\phi_p(\mathsf{s},\mathsf{o}))$
$\boldsymbol{r}_p$
$\star$
e_{s0} e_{s1} e_{s2} e_{o0} e_{o1} e_{o2}
subject
object
(b) HOLE

Figure 1: RESCAL and HOLE as neural networks. RESCAL represents pairs of entities via d^2 components (middle layer). In contrast, HOLE requires only d components.

A major appeal of TRANSE is that it requires very few parameters and moreover is easy to train. However, this simplicity comes also at the cost of modeling power. Wang et al. (2014) and Lin et al. (2015) proposed TRANSH and TRANSR respectively, to improve the performance of TRANSE on 1-to-N, N-to-1, and N-to-N relations. Unfortunately, these models lose the simplicity and efficiency of TRANSE.

Holographic Embeddings

In this section, we propose a novel compositional model for KGs and relational data. To combine the expressive power of the tensor product with the efficiency and simplicity of TRANSE, we use the circular correlation of vectors to represent pairs of entities, i.e., we use the compositional operator:

$$\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{a} \star \boldsymbol{b}, \tag{7}$$

where $\star : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ denotes *circular correlation*:[1]

$$[\boldsymbol{a} \star \boldsymbol{b}]_k = \sum_{i=0}^{d-1} a_i b_{(k+i) \bmod d}. \tag{8}$$

Hence, we model the probability of a triple as

$$\Pr(\phi_p(\mathsf{s},\mathsf{o}) = 1 | \Theta) = \sigma(\boldsymbol{r}_p^\top(\boldsymbol{e}_s \star \boldsymbol{e}_o)). \tag{9}$$

Due to its connections to holographic models of associative memory (which we will discuss in the next section) we refer to eq. (9) as *holographic embeddings* (HOLE) of KGs.

As a compositional operator, circular correlation can be interpreted as a compression of the tensor product. While the tensor product assigns a separate component $c_{ij} = a_i b_j$ for each pairwise interaction of entity features, in correlation each component corresponds to a *sum* over a fixed partition of pairwise interactions (see also fig. 2). Intuitively, a feature in the tuple representation is "on" if at least one partition of subject-object-interactions is on. This form of compression can be very effective since it allows to share weights in $\boldsymbol{r}_p$ for semantically similar interactions. For example, to model relational patterns in the *partyOf* relation, it might be sufficient to know whether subject and object are a *liberal person and liberal party* OR if they are a *conservative person and conservative party*. These interactions can then be grouped

[1]For notational convenience, we use zero-indexed vectors.

Table 1: (a) Memory complexity and runtime complexity for compositional representations with $\boldsymbol{e}_i \in \mathbb{R}^d$. (b) Memory complexity of embedding models.

(a) Compositional Representations

Operator	$\circ$	**Memory** $\boldsymbol{r}_p$	**Runtime** $\boldsymbol{r}_p^\top(\boldsymbol{e}_s \circ \boldsymbol{e}_o)$
Tensor Product	$\otimes$	$\mathcal{O}(d^2)$	$\mathcal{O}(d^2)$
Circular Correlation	$\star$	$\mathcal{O}(d)$	$\mathcal{O}(d \log d)$

(b) Embedding Models

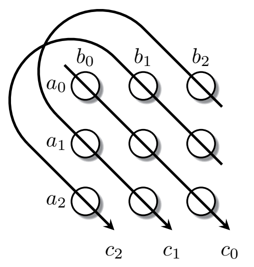
		on FB15k	
Method	**Memory Complexity**	d	Params
TRANSE	$\mathcal{O}(n_e d + n_r d)$	100	1.6M
TRANSR	$\mathcal{O}(n_e d + n_r d + n_r d^2)$	100	15.1M
ER-MLP	$\mathcal{O}(n_e d + n_r d + d_p d)$	200/200	3.3M
RESCAL	$\mathcal{O}(n_e d + n_r d^2)$	150	32.5M
HOLE	$\mathcal{O}(n_e d + n_r d)$	200	3.3M

in the same partition. Additionally, it is typically the case that only a subset of all *possible* interactions of latent features are relevant to model relational patterns. Irrelevant interactions can then be grouped in the same partitions and collectively be assigned a small weight in $\boldsymbol{r}_p$. Please note that the partitioning is not learned but fixed in advance through the correlation operator. This is possible because the entity representations are learned and the latent features can thus be "assigned" to the best partition during learning.

Compared to the tensor product, circular correlation has the important advantage that it does not increase the dimensionality of the composite representation (see also fig. 1b). The memory complexity of the tuple representation is therefore linear in the dimensionality d of the entity representations. Moreover, the runtime complexity is quasilinear (loglinear) in d, as circular correlation can be computed via

$$\boldsymbol{a} \star \boldsymbol{b} = \mathcal{F}^{-1}\left(\overline{\mathcal{F}(\boldsymbol{a})} \odot \mathcal{F}(\boldsymbol{b})\right)$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote the *fast Fourier transform* (FFT) and its inverse, $\bar{\boldsymbol{x}}$ denotes the complex conjugate of



$$\boldsymbol{c} = \boldsymbol{a} \star \boldsymbol{b}$$

$$c_0 = a_0 b_0 + a_1 b_1 + a_2 b_2$$
$$c_1 = a_0 b_2 + a_1 b_0 + a_2 b_1$$
$$c_2 = a_0 b_1 + a_1 b_2 + a_2 b_0$$

Figure 2: Circular correlation as compression of the tensor product. Arrows indicate summation patterns, nodes indicate elements in the tensor product. Adapted from (Plate 1995).

$\boldsymbol{x} \in \mathbb{C}^d$, and $\odot$ denotes the Hadamard (entrywise) product. The computational complexity of the FFT is $\mathcal{O}(d \log d)$. Table 1a summarizes the improvements of circular correlation over the tensor product. Table 1b lists the memory complexity of HOLE in comparison to other embedding models.

Circular convolution $* : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is an operation that is closely related to circular correlation and defined as

$$[\boldsymbol{a} * \boldsymbol{b}]_k = \sum_{i=0}^{d-1} a_i b_{(k-i) \bmod d}. \tag{10}$$

In comparison to convolution, correlation has two main advantages when used as a compositional operator:

Non Commutative Correlation, unlike convolution, is not commutative, i.e., $\boldsymbol{a} \star \boldsymbol{b} \neq \boldsymbol{b} \star \boldsymbol{a}$. Non-commutativity is necessary to model asymmetric relations (directed graphs) with compositional representations.

Similiarity Component In the correlation $\boldsymbol{a} \star \boldsymbol{b}$, a single component $[\boldsymbol{a} \star \boldsymbol{b}]_0 = \sum_i a_i b_i$ corresponds to the dot product $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$. The existence of such a component can be helpful to model relations in which the similarity of entities is important. No such component exists in the convolution $\boldsymbol{a} * \boldsymbol{b}$ (see also fig. 1 in the supplementary material).

To compute the representations for entities and relations, we minimize either eq. (2) or (3) via stochastic gradient descent (SGD). Let $\theta \in \{\boldsymbol{e}_i\}_{i=1}^{n_e} \cup \{\boldsymbol{r}_k\}_{k=1}^{n_r}$ denote the embedding of a single entity or relation and let $f_{spo} = \sigma(\boldsymbol{r}_p^\top (\boldsymbol{e}_s \star \boldsymbol{e}_o))$. The gradients of eq. (9) are then given by

$$\frac{\partial f_{spo}}{\partial \theta} = \frac{\partial f_{spo}}{\partial \eta_{spo}} \frac{\partial \eta_{spo}}{\partial \theta},$$

where

$$\frac{\partial \eta_{spo}}{\partial \boldsymbol{r}_p} = \boldsymbol{e}_s \star \boldsymbol{e}_o, \quad \frac{\partial \eta_{spo}}{\partial \boldsymbol{e}_s} = \boldsymbol{r}_p \star \boldsymbol{e}_o, \quad \frac{\partial \eta_{spo}}{\partial \boldsymbol{e}_o} = \boldsymbol{r}_p * \boldsymbol{e}_s. \tag{11}$$

The partial gradients in eq. (11) follow directly from

$$\boldsymbol{r}_p^\top (\boldsymbol{e}_s \star \boldsymbol{e}_o) = \boldsymbol{e}_s^\top (\boldsymbol{r}_p \star \boldsymbol{e}_o) = \boldsymbol{e}_o^\top (\boldsymbol{r}_p * \boldsymbol{e}_s) \tag{12}$$

and standard vector calculus. Equation (12) can be derived as follows: First we rewrite correlation in terms of convolution:

$$\boldsymbol{a} \star \boldsymbol{b} = \widetilde{\boldsymbol{a}} * \boldsymbol{b}$$

where $\widetilde{\boldsymbol{a}}$ denotes the *involution* of $\boldsymbol{a}$, meaning that $\widetilde{\boldsymbol{a}}$ is the mirror image of $\boldsymbol{a}$ such that $\widetilde{a}_i = a_{-i \bmod d}$ (Schönemann 1987, eq. 2.4). Equation (12) follows then from the following identities in convolution algebra (Plate 1995):

$$\boldsymbol{c}^\top (\widetilde{\boldsymbol{a}} * \boldsymbol{b}) = \boldsymbol{a}^\top (\widetilde{\boldsymbol{c}} * \boldsymbol{b}); \qquad \boldsymbol{c}^\top (\widetilde{\boldsymbol{a}} * \boldsymbol{b}) = \boldsymbol{b}^\top (\boldsymbol{a} * \boldsymbol{c}).$$

Similar to correlation, the circular convolution in eq. (11) can be computed efficiently via $\boldsymbol{a} * \boldsymbol{b} = \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{a}) \odot \mathcal{F}(\boldsymbol{b}))$.

Associative Memory

In this section we outline the connections of eq. (9) and eq. (11) to holographic models of associative memory. Such models employ a sequence of convolutions and correlations as used in holography to store and retrieve information (e.g., see (Gabor 1969; Poggio 1973)). In particular, holographic reduced representations (Plate 1995) *store* the association of $\boldsymbol{a}$ with $\boldsymbol{b}$ via their circular convolution

$$\boldsymbol{m} = \boldsymbol{a} * \boldsymbol{b},$$

and *retrieve* the item associated with $\boldsymbol{a}$ from $\boldsymbol{m}$ via

$$\boldsymbol{b}' \approx \boldsymbol{a} \star \boldsymbol{m} = \boldsymbol{b} * (\boldsymbol{a} \star \boldsymbol{a})$$

If $\boldsymbol{a} \star \boldsymbol{a} \approx \delta$ (the identity element of convolution), it holds that $\boldsymbol{b} \approx \boldsymbol{b}'$ and we can retrieve a noisy version of $\boldsymbol{b}$. For denoising, we can pass the retrieved vector through a clean-up memory, which returns the stored item with the highest similarity to the item retrieved from $\boldsymbol{m}$. For instance, if $\|\boldsymbol{a}\| = \|\boldsymbol{b}_i\| = 1$, we can perform the clean-up via

$$\boldsymbol{b} = \arg\max_{\boldsymbol{b}_i} \boldsymbol{b}_i^\top (\boldsymbol{a} \star \boldsymbol{m}) \tag{13}$$

Multiple elements are stored in $\boldsymbol{m}$ via superposition:

$$\boldsymbol{m} = \sum\nolimits_i \boldsymbol{a}_i * \boldsymbol{b}_i.$$

Hence, $\boldsymbol{m}$ acts in this scheme as a memory that stores associations between vectors which are stored and retrieved using circular convolution and correlation.

Consider now the following model of associative memory for relational data: Let $\mathcal{S}_o = \{(s, p) \mid \phi_p(\mathsf{s}, \mathsf{o}) = 1\}$ be the set of all subject-predicate indices for which the relation $\mathcal{R}_p(\mathsf{s}, \mathsf{o})$ is true. Next, we store these existing relations via convolution and superposition in the representation $\boldsymbol{e}_o$:

$$\boldsymbol{e}_o = \sum_{(s,p) \in \mathcal{S}_o} \boldsymbol{r}_p * \boldsymbol{e}_s \tag{14}$$

In this scheme, the compositional representation $\boldsymbol{e}_s \star \boldsymbol{e}_o$ of eq. (7) would be analogous to retrieving the stored predicates p that exist between s and o. Similarly, computing $\sigma(\boldsymbol{r}_p^\top (\boldsymbol{e}_s \star \boldsymbol{e}_o))$ as in eq. (9) is analogous to computing the probability that $\boldsymbol{r}_p$ is included in the retrieved relations, i.e., that we have seen the triple $\mathcal{R}_p(\mathsf{s}, \mathsf{o})$. The norm constraints of eq. (13) can either be enforced directly (by projection the embeddings onto the unit circle) or through the regularization of the embeddings (which is equivalent to $\|\boldsymbol{e}_i\| \leq C_e$, $\|\boldsymbol{r}_k\| \leq C_r$, where C_e, C_r depend on the regularization parameter).

An important difference of HOLE to associative memory is that it does not only memorize, but it generalizes in a well defined way: In associative memory we are given the embeddings and store the associations directly, typically via Hebbian learning (e.g., see eq. (14)). In HOLE, we do not simply store the associations, but instead learn the embeddings that best explain the observed data. By iterating over $\mathcal{D}$ with SGD, we update the embeddings of the objects via

$$\boldsymbol{e}_o^{t+1} \leftarrow \boldsymbol{e}_o^t - \mu \frac{\partial \mathcal{L}}{\partial f} \frac{\partial f}{\partial \eta} (\boldsymbol{r}_p^t * \boldsymbol{e}_s^t), \tag{15}$$

where μ denotes the learning rate. Please note that eq. (15) is analogous to the association of predicate and subject in holographic associative memory. Hence, we can interpret eq. (15) as adapting the "memory" $\boldsymbol{e}_o$, such that the retrieval of the observed facts is improved. The same analogy holds for the updates of $\boldsymbol{e}_s$ and $\boldsymbol{r}_p$, however with the roles of correlation and convolution in storage and retrieval reversed. Moreover, in minimizing eq. (2) via eq. (15), we are estimating a probability distribution over possible states of the knowledge graph which allows us to predict the probability of any *possible* triple in the graph (Nickel et al. 2015).

Experiments

Knowledge Graphs

To evaluate its performance for link prediction on knowledge graphs, we compared HOLE to state-of-the-art models on two commonly used benchmark datasets for this task:

WN18 WordNet is a KG that groups words into synonyms and provides lexical relationships between words. The WN18 dataset consists of a subset of WordNet, containing 40,943 entities, 18 relation types, and 151,442 triples.

FB15k Freebase is a large knowledge graph that stores general facts about the world (e.g., harvested from Wikipedia, MusicBrainz, etc.). The FB15k dataset consists of a subset of Freebase, containing 14,951 entities, 1345 relation types, and 592,213 triples.

For both datasets we used the fixed training-, validation-, and test-splits provided by Bordes et al. (2013). As baseline methods, we used RESCAL, TRANSE, TRANSR, and ER-MLP. To facilitate a fair comparison we reimplemented *all* models and used the identical loss and optimization method for training, i.e., SGD with AdaGrad (Duchi, Hazan, and Singer 2011) and the ranking loss of eq. (3). This improved the results of TRANSE and RESCAL significantly on both datasets compared to results reported by Bordes et al. (2013).[2]

Following Bordes et al. (2013), we generated negative relation instances for training by corrupting positive triples and used the following evaluation protocol: For each true triple $\mathcal{R}_p(\mathsf{s}, \mathsf{o})$ in the test set, we replace the subject s with each entity $\mathsf{s}' \in \mathcal{E}$, compute the score for $\mathcal{R}_p(\mathsf{s}', \mathsf{o})$, and rank all these instances by their scores in decreasing order. Since there can exist multiple true triples in the test set for a given predicate-object pair, we remove all instances from the ranking where $\mathcal{R}_p(\mathsf{s}', \mathsf{o}) = 1$ and $\mathsf{s} \neq \mathsf{s}'$, i.e., we consider only the ranking of the test instance among all wrong instances (which corresponds to the "Filtered" setting in Bordes et al. (2013)). We then repeat this procedure by replacing the object o. To measure the quality of the ranking, we use the mean reciprocal rank (MRR) which is commonly used in information retrieval and in contrast to mean rank is less sensitive to outliers. In addition to MRR, we report the ratio in which $\mathcal{R}_p(\mathsf{s}, \mathsf{o})$ occurs within the first n results (denoted by "Hits at n"). We optimized the hyperparameters of all models via extensive grid search and selected the model with the best filtered MRR score on the validation set. The results of these experiments are shown in table 2a. It can be seen that HOLE is able to outperform the considered baseline methods significantly and consistently on both datasets. For instance, TRANSE and TRANSR rank the test instance only in 11.5% and 33.5% of the cases as the most likely triple in WN18 (Hits at 1). In contrast, HOLE ranks the test instance in 93.0% of the cases as the most likely instance. While less pronounced, similar results can be observed on FB15k. In table 1b, we report the dimensionality d and the resulting number of parameters of the selected models. It can be seen that HOLE is far more efficient in the number of parameters compared to the tensor product model RESCAL. Although the dimensionality d of the HOLE embedding is larger than RESCAL's (what is to be expected due to the compressive effect of correlation), the overall number of parameters is significantly reduced as its memory complexity depends only linearly on d. Also, HOLE is typically very fast to compute. On standard hardware (Intel Core(TM) i7U 2.1GHz) and for $d = 150$ (as used in the experiments) the runtime to compute the probability of a single triple is around 40μs. To compute all embeddings, a single epoch on WN18 takes around 11s (earlier epochs are slower since more examples violate the margin). Typically, we need 200-500 epochs (depending on the dataset) to arrive at the best estimates for the embeddings.

Relational Learning

We have shown that HOLE can predict triples successfully in knowledge graphs. In additional experiments, we wanted to test the relational learning capabilities of the compositional representation. For this purpose, we used the countries dataset of Bouchard, Singh, and Trouillon (2015), which consists of 244 countries, 22 subregions (e.g., *Southern Africa*, *Western Europe*) and 5 regions (e.g., *Africa*, *Americas*). Each country is located in exactly one region and subregion, each subregion is located in exactly one region, and each country can have a number of other countries as neighbors. From the raw data we created a relational representation with two predicates: *locatedIn*(e_1, e_2) and *neighborOf*(e_1, e_2). The task in the experiment was to predict *locatedIn*(c, r) instances, where c ranges over all countries and r over all regions in the data. The evaluation protocol was the following: First, we split all countries randomly in train (80%), validation (10%), and test (10%) set, such that for each country in the test set there is at least one neighbor in the training set. Next, we removed triples from the test and validation set in three different settings:

S1) In the basic setting we only set *locatedIn*(c, r) to missing for countries in the test/valid. set. In this setting, the correct

[2] TRANSE in its original implementation used SGD without AdaGrad. RESCAL used the least-squares loss and ALS updates.

Table 2: Results for link prediction on WordNet (WN18), Freebase (FB15k) and Countries data.

(a)

	WN18					FB15k				
	MRR		Hits at			MRR		Hits at		
Method	Filter	Raw	1	3	10	Filter	Raw	1	3	10
TRANSE	0.495	0.351	11.3	88.8	94.3	0.463	0.222	29.7	57.8	**74.9**
TRANSR	0.605	0.427	33.5	87.6	94.0	0.346	0.198	21.8	40.4	58.2
ER-MLP	0.712	0.528	62.6	77.5	86.3	0.288	0.155	17.3	31.7	50.1
RESCAL	0.890	0.603	84.2	90.4	92.8	0.354	0.189	23.5	40.9	58.7
HOLE	**0.938**	**0.616**	**93.0**	**94.5**	**94.9**	**0.524**	**0.232**	**40.2**	**61.3**	73.9

(b)

	Countries		
	AUC-PR		
Method	S1	S2	S3
Random	0.323	0.323	0.323
Frequency	0.323	0.323	0.308
ER-MLP	0.960	0.734	0.652
RESCAL	**0.997**	0.745	0.650
HOLE	**0.997**	**0.772**	**0.697**

relations can be predicted from patterns of the form:

$$locatedIn(c, s) \wedge locatedIn(s, r) \Rightarrow locatedIn(c, r)$$

where s refers to the country's subregion.

S2) In addition to the triples of S1, we set *locatedIn*(c, s) to missing for all countries c in the test/valid. set and all subregions s in the data. In this setting, the correct triples can be predicted from:

$$neighborOf(c_1, c_2) \wedge locatedIn(c_2, r) \Rightarrow locatedIn(c_1, r)$$

This is a harder task than S1, since a country can have multiple neighbors and these can be in different regions.

S3) In addition to the triples of S1 and S2 we set *locatedIn*(n, r) to missing for all neighbors n of all countries in the test/valid. set and all regions r in the data. In this setting, the correct triples can be predicted from:

$$neighborOf(c_1, c_2) \wedge locatedIn(c_2, s) \wedge locatedIn(s, r) \Rightarrow locatedIn(c_1, r)$$

This is the most difficult task, as it not only involves the *neighborOf* relation, but also a path of length 3.

See fig. 3 for an illustration of the data structure and the test settings. We measured the prediction quality via the area under the precision-recall curve (AUC-PR). The results of the experiments are shown in table 2b. It can be seen that HOLE is very successful in these learning tasks. For S1, the missing triples are predicted nearly perfectly. Moreover, even for the most difficult task S3, HOLE achieves very good results, especially since not every country's region can be predicted from its neighbors (e.g., islands have no neighbors). The poorer results of RESCAL and ER-MLP can likely be explained with overfitting (although the models are regularized), since the difference to HOLE is reduced when the hyperparameters are optimized on the test set instead of the validation set. We observed similar results as in this experiment on commonly used benchmark datasets for statistical relational learning. Due to space constraints, we report these experiments in the supplementary material.

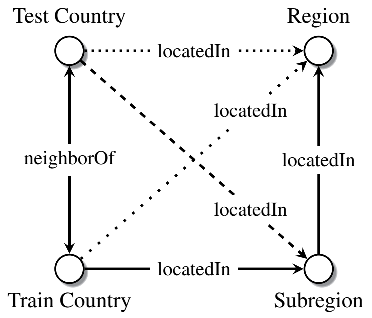


Figure 3: Removed edges in countries experiment: S1) dotted S2) dotted and dashed S3) dotted, dashed and loosely dotted.

Conclusion and Future Work

In this work we proposed HOLE, a compositional vector space model for knowledge graphs that is based on the circular correlation of vectors. An attractive property of circular correlation in this context is that it creates fixed-width representations, meaning that the compositional representation has the same dimensionality as the representation of its constituents. In HOLE, we exploited this property to create a compositional model that can capture rich interactions in relational data but simultaneously remains efficient to compute, easy to train, and very scalable. Experimentally we showed that HOLE provides state-of-the-art performance on a variety of benchmark datasets and that it can model complex relational patterns while being very economical in the number of its parameters. Moreover, we highlighted connections of HOLE to holographic models of associative memory and discussed how it can be interpreted in this context. This creates not only a link between relational learning and associative memory, but also allows for principled ways to query the model, for instance in question answering. In future work we plan to further exploit the fixed-width representations of holographic embeddings in complex scenarios, since they are especially suitable to model higher-arity relations (e.g., *taughtAt(John, AI, MIT)*) and facts about facts (e.g., *believes(John, loves(Tom, Mary))*).

Acknowledgments & Reproducibility

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216. The code for models and experiments used in this paper is available at https://github.com/mnick/holographic-embeddings.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*. Springer. 722–735.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1247–1250.

Bordes, A.; Weston, J.; Collobert, R.; and Bengio, Y. 2011. Learning Structured Embeddings of Knowledge Bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Advances in Neural Information Processing Systems*, 2787–2795.

Bouchard, G.; Singh, S.; and Trouillon, T. 2015. On Approximate Reasoning Capabilities of Low-Rank Vector Spaces. In *2015 AAAI Spring Symposium Series*.

Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. 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*, 601–610. ACM.

Dong, X. L.; Gabrilovich, E.; Murphy, K.; Dang, V.; Horn, W.; Lugaresi, C.; Sun, S.; and Zhang, W. 2015. Knowledge-based Trust: Estimating the Trustworthiness of Web Sources. *Proceedings of the VLDB Endowment* 8(9):938–949.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *The Journal of Machine Learning Research* 12:2121–2159.

Gabor, D. 1969. Associative Holographic Memories. *IBM Journal of Research and Development* 13(2):156–159.

Gentner, D. 1983. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science* 7(2):155–170.

Getoor, L., and Taskar, B. 2007. *Introduction to Statistical Relational Learning*. The MIT Press.

Guu, K.; Miller, J.; and Liang, P. 2015. Traversing Knowledge Graphs in Vector Space. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Kemp, C.; Tenenbaum, J. B.; Griffiths, T. L.; Yamada, T.; and Ueda, N. 2006. Learning Systems of Concepts with an Infinite Relational Model. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 381–388.

Krompaß, D.; Nickel, M.; and Tresp, V. 2014. Querying Factorized Probabilistic Triple Databases. In *The Semantic Web–ISWC 2014*. Springer Int. Publishing. 114–129.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Mitchell, J., and Lapata, M. 2008. Vector-based Models of Semantic Composition. In *Proceedings of ACL-08: HLT*, 236–244.

Muggleton, S. 1991. Inductive Logic Programming. *New generation computing* 8(4):295–318.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2015. A Review of Relational Machine Learning for Knowledge Graphs. *To appear in Proceedings of the IEEE*. arXiv:1503.00759.

Nickel, M.; Jiang, X.; and Tresp, V. 2014. Reducing the Rank in Relational Factorization Models by Including Observable Patterns. In *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc. 1179–1187.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *Proceedings of the 28th International Conference on Machine Learning*, 809–816.

Plate, T. 1995. Holographic Reduced Representations. *IEEE Transactions on Neural Networks* 6(3):623–641.

Poggio, T. 1973. On Holographic Models of Memory. *Kybernetik* 12(4):237–238.

Richardson, M., and Domingos, P. 2006. Markov Logic Networks. *Machine learning* 62(1-2):107–136.

Schönemann, P. 1987. Some algebraic relations between involutions, convolutions, and correlations, with applications to holographic memories. *Biological Cybernetics* 56(5-6):367–374.

Smolensky, P. 1990. Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems. *Artificial Intelligence* 46(1):159–216.

Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic Compositionality through Recursive Matrix-Vector Spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1201–1211.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. Y. 2013. Reasoning with Neural Tensor Networks for Knowledge Base Completion. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc. 926–934.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, 697–706.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Xu, Z.; Tresp, V.; Yu, K.; and Kriegel, H. 2006. Infinite Hidden Relational Models. In *In Proceedings of the 22nd International Conference on Uncertainity in Artificial Intelligence*.

Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*.