Representing Joint Hierarchies with Box Embeddings

Anonymous authors

Abstract

Learning representations for hierarchical and multi-relational knowledge has emerged as an active area of research. Box Embeddings [Vilnis et al., 2018, Li et al., 2019] represent concepts with hyperrectangles in $n$-dimensional space and are shown to be capable of modeling tree-like structures efficiently by training on a large subset of the transitive closure of the WordNet hypernym graph. In this work, we evaluate the capability of box embeddings to learn the transitive closure of a tree-like hierarchical relation graph with far fewer edges from the transitive closure. Box embeddings are not restricted to tree-like structures, however, and we demonstrate this by modeling the WordNet meronym graph, where nodes may have multiple parents. We further propose a method for modeling multiple relations jointly in a single embedding space using box embeddings. In all cases, our proposed method outperforms or is at par with all other embedding methods.

1. Introduction

Hierarchical relations, particularly hypernymy, are inherently present in natural language and useful in many common tasks. For example, in SNLI the sentence “A soccer game with multiple males playing” can be understood to entail “Some men are playing a sport” via the hypernym relation between “soccer” and “sport”. Question answering and language modeling in general can benefit from understanding these relationships, as a word can often be replaced by it’s hypernym and still yield a valid sentence or provide supporting evidence to answer a question. Vector embedding methods, where an entity is associated with a vector in $\mathbb{R}^n$, often use symmetric measures such as the dot product between these representations which cannot capture inherently asymmetric relations such as hypernymy.

In order to solve this problem, [Vendrov et al., 2016] introduced an asymmetric measure on vectors in the form of entailment cones in $\mathbb{R}_+^n$. [Vilnis et al., 2018, Li et al., 2019] generalized this notion to propose box embeddings which are hyperrectangles in $\mathbb{R}^n$. We build upon this work by proposing a new regularization loss for the box embedding model. This provides a weak scale on the embedding space resulting a significant performance improvement.
In this work, we explore the extent to which the edges from the transitive closure are necessary to model the tree-like large tree-like graphs, e.g., WordNet’s hypernymy. We further explore the meronymy relation from WordNet, which differs qualitatively from the hypernymy graph in that it is not as tree-like, comprised of many connected components where some nodes have multiple parents while others have no parents. We not only achieve state-of-the-art performance with our box embedding based method over other baselines when 10, 25, and 50% of the transitive closure edges are supplied during training but, most importantly, we outperform other methods by a significant margin when the training is being restricted to the transitive reduction only (0% transitive closure edges).

Finally, we introduce a method of modeling these two relations in the same space by introducing two representations for each entity. Our method is not specific to box embeddings, but rather can be thought of as a graph augmentation procedure which is general enough to allow for modeling with all existing baselines. In this multi-relational embedding settings we also compare against knowledge base completion baselines and demonstrate that box embeddings can outperform TransE [Bordes et al., 2013] and ComplEx [Trouillon et al., 2016] in this setting as well.

2. Related Work

As briefly mentioned in the introduction, Vendrov et al. [2016] provides an asymmetric measure on vectors in the form of the reversed product order on $\mathbb{R}^n_+$. This amounts to associating with each entity a cone with apex $x$, that is the set $\{y : \land_{i=1}^n y_i \geq x_i\}$. Lai and Hockenmaier [2017] provided a probabilistic interpretation of order embeddings (POE) by using the negative exponential measure or, equivalently, representing concepts as a cone in $[0,1]^n$. With this parameterization, the volume of a cone could be interpreted as the marginal (resp. joint) probability of the entity (resp. intersection) which it represents. This allowed for training and evaluation using conditional probabilities which mapped very naturally onto hypernym relations, eg. $P(\text{mammal}|\text{dog}) = 1$. However, representing entities probabilistic cones has a serious deficit – it cannot represent negative correlation, that the case when $P(X|Y) \geq P(X)$.

Box embeddings, introduced in Vilnis et al. [2018], solves this issue by introducing another vector, effectively associating each entity with an $n$-dimensional hyper-rectangle. Their method have been shown to effectively model hypernymy by training on a large subset of the transitive closure of the WordNet hypernym graph.

Learning hierarchical representations of symbolic data by embedding them into hyperbolic space has also been one of the highly researched area related to this work. Nickel and Kiela [2017], Chamberlain et al., 2017 demonstrate the effectiveness of $n$-dimensional Poincaré ball to model tree structured data. Ganea et al. [2018] further extend this idea Hyperbolic entailment cones that can be thought of as the hyperbolic analog to order embeddings.
3. Method

We first present box embeddings as they were introduced in [Vilnis et al., 2018], briefly touch on the “soft” volume calculation presented in [Li et al., 2019], and then discuss some learning adaptations present in our version of the model.

3.1 Representation

The box-space [Vilnis et al., 2018] is the set of all $n$-dimensional hyperrectangles. Formally, if $\mathbb{B} = \{[a, b] \mid a, b \in \mathbb{R}, \ b \geq a\}$ is the set of all closed intervals of $\mathbb{R}$, then an $n$-dimensional box-space is $\mathbb{B}^n$. Just as in vector embedding models where each entity is represented by a vector in the euclidean space, in box-space each entity $e$ is represented by a box $\alpha_e \in \mathbb{B}^n$ as shown in figure 1. We parametrize a box using the minimum and maximum coordinates in each dimension, $(\alpha_m, \alpha_M)$, where $\alpha_m, \alpha_M \in \mathbb{R}^n$ and $\alpha_{m,i} \leq \alpha_{M,i}$ for each $i \in \{1, \ldots, n\}$. Boxes provides a natural way to embed a partial order using containment.

![Figure 1: An element of box-space.](image)

3.2 Probabilistic Interpretation

In addition to the natural containment operation, we can consider the volume of boxes. If we normalize the space to have volume 1 (by dividing by the smallest containing box, say) the boxes can be interpreted as a parameterization of a joint probability distribution over binary random variables, where the marginal probability of a given entity is given by the volume of it’s box. Intersections of boxes yield either another box or the empty set, and in either case we can compute the volume of the intersection as

$$\text{Vol}(\alpha \cap \beta) := \prod \max(\min(\alpha_{M,i}, \beta_{M,i}) - \max(\alpha_{m,i}, \beta_{m,i}), 0).$$

(1)

We can interpret this volume as the joint probability for concepts $\alpha$ and $\beta$, and furthermore calculate (for example) $^1$

$$\Pr(\alpha | \beta) = \frac{\text{Vol}(\alpha \cap \beta)}{\text{Vol}(\beta)}.$$ 

(2)

[Vilnis et al. 2018] uses these conditional probabilities to model the hypernym graph of WordNet. For example, given an edge from MAN to PERSON (indicating that a man is a

---

1. Note that when calculating conditional probabilities it is not necessary to normalize the volume of the space to size 1.
person) we first convert this to a conditional probability between binary random variables, \( \Pr(\text{PERSON} \mid \text{MAN}) = 1 \), which is then trained using binary cross-entropy loss and stochastic gradient descent. Figure 2 shows the corresponding boxes capturing this edge.

![Diagram](image.png)

Figure 2: Formulation of conditional probability using intersection of boxes. Note that in this figure \( \text{Man} \cap \text{Person} = \text{Man} \).

### 3.3 Softplus Volume

Boxes which should overlap may become disjoint, either due to initialization or during training, and the currently described model has no training signal in this instance. Vilnis et al. [2018] made use of a surrogate loss function to handle this case, however Li et al. [2019] introduced a new method which replaces the ReLU from (1) with the softplus function, \( \text{softplus}_t(x) = t \log(1 + e^{x/t}) \), which smoothed the loss landscape and resulted in easier training. The volume calculation under this approximation becomes

\[
\text{Vol}(\alpha \cap \beta) := \prod \text{softplus}_t(\min(\alpha_{M,i}, \beta_{M,i}) - \max(\alpha_{m,i}, \beta_{m,i})). \tag{3}
\]

We note that (3) is always positive, which means all boxes “intersect” one another under this volume calculation. The temperature parameter \( t \) can be tuned, and as \( t \to 0 \) we recover the original box model. We will exclusively use this soft volume calculation when computing box volumes, and in our experiments we always select the temperature based on dev set performance.

### 3.4 Absence of explicit probability labels and use of regularization

Previous work on box embeddings [Vilnis et al., 2018] uses pre-computed unary marginals \( \Pr(A) \) and pairwise conditionals \( \Pr(A \mid B) \) as labels while training the box embedding model. These marginal probabilities were calculated using the graph structure - leaves were given marginal probability equal to \( 1/\#\text{nodes} \), and parents were assigned marginal probability equal to the sum of their children plus \( 1/\#\text{nodes} \). How “tight” the parent boxes are to their children can be seen as a form of regularization. In general, setting marginal probabilities can help to set a scale for the overall model.

For our model, however, we avoid pre-computing marginal probabilities and simply use the conditional probability targets with binary cross-entropy loss and augment it with an
additional regularization loss which can be written as

\[ L = \sum_i^N -y_i \log p_i - (1 - y_i) \log(1 - p_i) + \sum_j^{N_e} \mathbb{I}_{[\text{Vol}(\alpha(j)) > \tau]} \text{Vol}(\alpha(j)), \]  

\[ (4) \]

\( N \) being the total number of examples (including the random negatives), \( N_e \) the total number of entities, \( y_i \) the label, \( p_i \) the conditional probability calculated using equation 2, and \( \mathbb{I} \) the indicator function.

Without any global scale i.e., the marginal probabilities, we get poor training results, so we introduce this regularization measure by penalizing the size of boxes when they become greater than a fixed volume. This provides a weak scale to the boxes in the embedding space and results in a significant performance improvement over unregularized training without the need to precompute unary marginals.

### 3.5 Modeling joint hierarchy

Inter-relational semantics implicit in multi-relational knowledge graphs can be used to learn coherent representations for the entities present in the graph. In particular, the relations discussed in this work, i.e., the \texttt{IsA} and \texttt{HasPart} relations should obey the following rules.

\[ \texttt{IsA}(a, b) \land \texttt{HasPart}(b, c) \Rightarrow \texttt{HasPart}(a, c) \]  

\[ (5) \]

\[ \texttt{IsA}(a, b) \land \texttt{HasPart}(c, a) \Rightarrow \texttt{HasPart}(c, b) \]  

\[ (6) \]

For instance, every \texttt{Man} \texttt{IsA} \texttt{Person} and every \texttt{Person} \texttt{HasPart} \texttt{Eye}. Hence, every \texttt{Man} \texttt{HasPart} \texttt{Eye}. Similarly, \texttt{Eye} \texttt{IsA} \texttt{Organ} and \texttt{Person} \texttt{HasPart} \texttt{Eye} implies Person
HasPart Organ. Figure 4 shows the subgraph describing this example. The red edges and the dotted red edges are implied HasPart edges which might be missing in the original knowledge graph. We show that boxes can model these two relations in a single space by using two boxes per node while preserving the semantics described above.

For each node $x$ we embed two boxes, one which represents “Things which are $x$” and one which represents “Things which have $x$ as a part”. This is depicted in figure 3, where the entity “Man” is represented by the green Man box and the red HasPart-Man box. The edges 3,4,5 and 7, in figure 4 are captured by the inclusion of Man and Person boxes inside the HasPart-Eye and HasPart-Organ boxes, while the edges 1 and 2 are captured by the inclusion of Man inside the Person Box, and Eye inside the Organ box, respectively.

This form of modeling multiple relations using augmentation is not limited to the box embedding model. The main idea is to create a new graph with twice as many nodes and edges from the IsA hierarchy, with additional PartOf edges so as to make the semantics coherent. Any model capable of modeling a DAG can be evaluated on it’s ability to model this graph.

4. Experiments

In order to demonstrate the efficacy of the box embeddings model on hierarchical structured data, we evaluate on the hypernymy and meronymy relations of WordNet. Following the recent works of [Nickel and Kiela, 2017, Ganea et al., 2018], we train on the transitive reduction, using increasing amounts of edges from the transitive closure of the those lexical relations. We pose the problem as a binary classification task on the unseen test edges which include the remaining edges from the transitive closure along with a fixed set of random negatives for the respective relations.

4.1 Datasets

1. Hypernymy (IsA)

For hypernymy, we used the dataset from [Ganea et al., 2018]. The WordNet noun hypernymy dataset contains 82,114 nodes (omitting the root). The transitive reduction of the hypernymy hierarchy is used for training. Additionally, there are training datasets with 10%, 25%, and 50% of the transitive closure edges. For dev and test datasets, 10% of the edges (5% each) are sampled from the rest of the transitive closure. For each test and validation edge, a fixed set of negative samples of size 10 was generated.

2. Meronymy (HasPart)

We created a meronomy dataset in the exact same way. Out of the 82,114 nodes in WordNet only 11,235 of them have meronym relations associated with them. We note that this graph is not nearly as tree-like - there are 2,083 connected components, 2,860 nodes with no parents and 1,013 nodes with multiple parents. Again, following [Ganea et al., 2018], we start with the transitive reduction and add 0%, 10%, 25%, and 50% of the transitive closure to the training data in order to observe corresponding improvement in the performance of the model. Dev and test datasets were created in
Table 1: Details of the hypernymy and meronymy hierarchies. This training set corresponds to the 0% transitive closure setting.

<table>
<thead>
<tr>
<th></th>
<th>Transitive Closure</th>
<th>Transitive Reduction</th>
<th>Validation (pos/neg)</th>
<th>Test (pos/neg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypernym</td>
<td>84363</td>
<td>661127</td>
<td>28838/288380</td>
<td>28838/288380</td>
</tr>
<tr>
<td>Meronym</td>
<td>9678</td>
<td>30333</td>
<td>5164/51640</td>
<td>5164/51640</td>
</tr>
</tbody>
</table>

the same way as for hypernymy. Please refer to Table 1 for the details of the dataset.

3. Joint Hierarchy ($\text{IsA}$ and $\text{HasPart}$)

For the joint hierarchy, we start with the transitive reduction of meronymy and hypernymy, and create a new graph as described in section 3.5. Specifically, we add the implied $\text{HasPart}$ edges (shown as the red arrows in figure 4). The test data is comprised of a subset of the implied edges (shown as dotted red in figure 4) which are of the form

$$\underbrace{\text{IsA} \circ \text{IsA} \cdots \text{IsA}}_{m \text{ times}} \circ \text{HasPart} \circ \underbrace{\text{IsA} \circ \text{IsA} \cdots \text{IsA}}_{n \text{ times}} = \text{HasPart}. \quad (7)$$

The dev and test created in this manner have 94807 and 94806 positive edges respectively which include all $\text{HasPart}$ edges as described in (7) for all $m, n \in \{1, 2\}$.

4.2 Baselines

We compared our model with the recent strong baseline models such as:

1. Order Embeddings [Vendrov et al., 2016]: As mentioned in the introduction, this method embeds entities as cones in $\mathbb{R}^d$. Representationally, order embeddings are equivalent to box embeddings where the min coordinate is fixed to the origin, however learning differs in that the objective function optimizes an order-violation penalty. (See [Vendrov et al., 2016] for more details.)

2. Poincaré Embeddings [Nickel and Kiela, 2017]: In this work, the authors learn embeddings in an $n$-dimensional Poincaré ball which naturally captures the tree-like structure via the constant negative curvature of hyperbolic geometry. This work uses a symmetric distance function which is not ideal for asymmetric relational data, however. We use the same heuristics followed by [Ganea et al., 2018] to mitigate this problem.

3. Hyperbolic Entailment Cones [Ganea et al., 2018]: This work models hierarchical relations as partial orders defined using a family of nested geodesically convex cones in hyperbolic space.

4. TransE and ComplEx [Bordes et al., 2013, Trouillon et al., 2016]: The joint hierarchy can be also considered as multi-relational data with two relations. We use the most
popular translational distance based and matrix factorization based models, TransE and ComplEx respectively.

In order to show the effectiveness of the proposed regularisation, we include the performance of the vanilla box embeddings [Li et al., 2019] as well.

4.3 Training details

In all our experiments, we keep the embedding dimension as 10 for all the baseline models, while for our model we use embedding dimension as 5. This results in having 10 parameters per node and ensures a fair comparison. For the task of learning the joint hierarchy with multi-relation prediction models like TransE and ComplEx, we keep their embedding dimension as 20 to account for the doubling of parameters due the introduction of two embeddings per node in Order embeddings, Poincaré embeddings, Hyperbolic entailment cones and our model. The effect of increasing the embedding dimensions is reported in Appendix B.

We use the validation set to find the best threshold for the classification and obtain the F1 score on the test set using this threshold. Our best performing models have softbox temperature of between 0.2 and 0.5. We obtain the best learning rate for every model using random search and performance the validation set.

5. Results

The F1 scores for the hypernymy and meronymy predictions are presented in table 2. We note that our box embedding method outperforms all baselines by a large margin in the single hierarchy settings. Particularly, in the transitive reduction (0%) setting for hypernymy, we achieve 40% relative improvement compared to the most competitive baseline (order embeddings). This demonstrates the capability of box embeddings to reconstruct the whole transitive closure graph given only the transitive reduction. We observe a similar trend when modeling meronymy. When transitive closure edges are added to the training data (10%, 25%, 50%), the gap between our method and the baselines closes, however the former still remains superior.

In Table 3, we report the F1 scores for the test edges of the joint hierarchy, which are the composite edges created by the rules (7) mentioned in section 4.1. We outperform the multi-relation knowledge base embedding methods as well as hyperbolic models (Poincaré and hyperbolic entailment cones). While methods like TransE and ComplEx lack the inductive bias required to model such highly connected relations, the hyperbolic space methods struggle due to their inability to train very well. Order embeddings, on the other hand, slightly outperform our model, but the latter is essentially at par while also achieving superior performance on the task of modeling single hierarchy. The high performance of order embeddings on this task can be attributed to the “local” nature of the evaluation (we traverse only 2 steps up and down the hypernym graph by keeping \(m, n \in \{1, 2\}\) in eq. 7) which might not be exposing the representation limitations of order embeddings – something that is exposed while modeling the relations individually.
Table 2: Test F1 scores of various methods for predicting the transitive closure of WordNet’s hypernym meronym relations when training on increasing amounts of edges from the transitive closure. Baselines for hypernymy are as-reported in [Ganea et al., 2018].

<table>
<thead>
<tr>
<th>Transitive Closure Edges</th>
<th>Hypernym</th>
<th>Meronym</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
<td>10%</td>
</tr>
<tr>
<td>Order Embedding</td>
<td>43.0%</td>
<td>69.7%</td>
</tr>
<tr>
<td>Poincaré Embedding</td>
<td>28.9%</td>
<td>71.4%</td>
</tr>
<tr>
<td>Hyperbolic Entailment Cones</td>
<td>32.2%</td>
<td>85.9%</td>
</tr>
<tr>
<td>Box Embeddings (w/o regularization)</td>
<td>45.4%</td>
<td>72.6%</td>
</tr>
<tr>
<td>Box Embeddings (Our Method)</td>
<td><strong>60.2%</strong></td>
<td><strong>90.0%</strong></td>
</tr>
</tbody>
</table>

Table 3: Test F1 scores of various methods for predicting the implied HasPart edges

<table>
<thead>
<tr>
<th>Embedding Model</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poincaré Embeddings</td>
<td>43.80%</td>
</tr>
<tr>
<td>Hyperbolic Entailment Cones</td>
<td>44.00%</td>
</tr>
<tr>
<td>TransE</td>
<td>57%</td>
</tr>
<tr>
<td>ComplEx</td>
<td>60.61%</td>
</tr>
<tr>
<td>Order Embeddings</td>
<td><strong>68.50%</strong></td>
</tr>
<tr>
<td>Box embeddings</td>
<td>68.10%</td>
</tr>
</tbody>
</table>

5.1 Qualitative analysis

In order to visualize the effect of using “soft” boxes, we train a 2-dimensional box embedding model on the hypernym relation of WordNet. Due to the softening of the boxes, the model does not require the dog.n.01 box to be completely contained inside the domestic_animal.n.01 box to produce high score for the edge between them. Figure 5 shows the visualization obtained by plotting the hard edges using the minimum and maximum coordinates of the soft boxes. For the Joint Hierarchy model, we obtain similar visualizations which are given in appendix A.

In order to model a hierarchy, the embedding model has to accommodate the exponential growth in the number of nodes while moving down the hierarchy. The loss function for box embedding model encourages the embedding of the head node to contain the embedding of the tail node. Hence the number of embeddings having low volume should be high. This intuition is confirmed by figure 6.

6. Conclusion

In this paper we have explored the capability of box embeddings to model hierarchical data. We demonstrated that it provides superior performance compared to alternative methods and requires less of the transitive closure. We also demonstrated that boxes are capable of learning data which is less tree-like, and introduced a method of embedding a joint hierarchy in a single space by augmenting the graph. There are several promising directions which can be explored in future work. It would be of interest to expand the number of relations...
Figure 5: Visualization of 2-dimensional boxes for a subset of entities from the WordNet hypernym hierarchy after training on the entire WordNet reduction.

which are modeled in a single box space and to extend the method to model non-transitive relations. Also, the knowledge encoded in these embeddings can be exploited by using them as a representation layer in neural network models for various downstream tasks like question answering, natural language inference, etc.
Figure 6: Histogram of the log-volume of 5-dimensional box embeddings for the hypernym relation of WordNet. Here the count (y-axis) is also in log scale.

Acknowledgments

Omitted for blind review.

References


Appendix A. Joint Hierarchy Visualization

We qualitatively validate the performance of our box embedding method on modelling the IsA-HasPart joint hierarchy by visualizing the box embeddings for subsets of the joint hierarchy. One such subset occurring in WordNet is shown in figure 7. The box embeddings corresponding to this subset are shown in figures 8 and 9. It is worth noting that even with a classification F1 score of 68%, the visualizations validate our intuition.

Figure 7: An example of joint hierarchy occurring in the WordNet (in the IsA and PartOf relations)

The first parts of figures 8 and 9 show how the box embeddings are able to represent the implied HasPart edges – the inclusion of the Car and Sedan boxes inside the HasPart Car Door, HasPart Door and HasPart Movable Barrier boxes capture the edges 3, 4 and 5 in figure 7. On the other hand, the second parts of the same figures show that, by inclusion of Sedan inside the Car box and Car Door inside the Door box, that our model still retains the edge information for edges 1 and 2.
Figure 8: A 2-dimensional projection (on first two dimensions) of the 5-dimensional box embedding model trained to model the joint hierarchy. The embeddings plotted here correspond to the entities shown in figure 7. The two pictures show box embeddings in the same space but are separated into two plots for ease of visualization and comprehension. Note, some of the larger boxes extending outside of the plot range are not shown completely.
Figure 9: A 2-dimensional projection (on dimensions 3 and 4) of the 5-dimensional box embedding model trained to model the joint hierarchy. The embeddings plotted here correspond to the entities shown in figure 7. The two pictures show box embeddings in the same space but are separated into two plots for ease of visualization and comprehension. Note, some of the larger boxes extending outside of the plot range are not shown completely.
Appendix B. Effect of dimensions

We also investigate how different models scale with the number of parameters. As shown in table 4, the performance of all the models increase rapidly with the number of parameters, but plateaus after a certain point. This shows that all the models including box embeddings have strong inductive bias for modelling hierarchies. That is, they can model large hierarchical graphs with a small number of parameters.

<table>
<thead>
<tr>
<th>Model</th>
<th>Test F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector Dimension (parameters per node)</td>
<td>10  20  50  100</td>
</tr>
<tr>
<td>Order Embedding</td>
<td>43  49.2  52.2  53</td>
</tr>
<tr>
<td>Poincare</td>
<td>28.9  30.9  31.2  31.4</td>
</tr>
<tr>
<td>Hyperbolic entailment cones</td>
<td>32.2  33.2  38.5  39.4</td>
</tr>
<tr>
<td>Box Embedding (our method)</td>
<td>60  65  67  68</td>
</tr>
</tbody>
</table>

Table 4: Effect of increasing model dimensions on the test F1 scores for the task of learning the Hypernym relation (with 0% transitive closure edges in the training set).