Training Box Embeddings for Joint Probabilities

Anonymous authors

Abstract

This paper describes experiments training box embeddings to learn a set of N-way joint probability distributions over a set of category labels, where the categories are arranged in an expert-defined taxonomy. The sample space is PubMed abstracts with human assigned MeSH subject labels. Since the space of all possible joint probabilities is too large, we explore the efficacy of box embeddings with different training regimes on learning the joint probability distributions between the MeSH categories. The main purpose of this paper is to share experimental results and experiences with a new embedding method that promises more expressiveness, but with which the community has little experience. We show that training with samples from the space of joint probabilities is infeasable, and that training with conditional probabilities is much faster than with the more intuitive approach of using instance boxes, which do not scale well. Conditional probabilities require fewer parameters and less training data, with equivalent to slightly-better performance in reproducing N-way joint probabilities (up to N=9).

1. Introduction

Box embeddings were introduced in [Vilnis et al., 2018] and provide a clear expressiveness increase over point embeddings: in addition to similarity measures like distance, boxes can represent containment, overlap, and negation. Further, boxes are closed under intersection, making them a very natural representation for joint and conditional probabilities. These properties may mean box embeddings are ideally suited to representing more continuous-valued knowledge graphs (as opposed to discrete valued), however the community has little practical experience with them. The primary purpose of this paper is to add to that body of community experience.

The first experiments with boxes used their expressive power for encoding taxonomies, in this paper we experiment with using boxes for encoding joint probabilities. In general we look at the problem of encoding a large set of N-way joint probabilities over a set of categories. The space of possible joint probabilities is too large to enumerate, so we investigate two sampling methods and evaluate their efficacy for training boxes to generalize to the full distribution.
2. Problem Definition

We start with a graph $G$, whose nodes represent MeSH categories (e.g. “Asthma”) and whose edges represent the taxonomic relation. The edges are directional and form a DAG and a partial order on the nodes in $G$, as one would expect in a knowledge-base, e.g. $\langle \text{Asthma}, \text{BronchialDiseases} \rangle$ and $\langle \text{BronchialDiseases}, \text{RespTractDiseases} \rangle$ indicates a progression “up” the taxonomy from more specific to more general labels, and the community expects the transitive closure to hold so that $\langle \text{Asthma}, \text{RespTractDiseases} \rangle$ is implied by the previous two edges.

We now add a set of nodes to $G$ denoting the articles in PubMed, and a new edge type denoting the instance or type relation between MeSH categories and PubMed articles. We distinguish the new edge type ($R_I$) from the taxonomic one ($R_{SC}$) and note that $R_I$ forms a bipartite graph that partitions instances (PubMed articles, $I$) and categories (MeSH terms, $C$): $R_I : I \times C$, while $R_{SC} : C \times C$.

In practice, a PubMed article will have many subject labels, and $R_I$ is expected to be transitive over $R_{SC}$, in what used to be called inheritance, i.e.: $R_I = \{ \langle i_1, c_1 \rangle \}$ and $R_{SC} = \{ \langle c_1, c_2 \rangle \} \implies R'_I = \{ \langle i_1, c_1 \rangle, \langle i_1, c_2 \rangle \}$. Similarly we denote the transitive closure of subcategory edges as $R'_{SC}$.

This formulation is meant to generalize to any knowledge-based problem for which instances can be assigned multiple categories and those categories are arranged in a taxonomy, e.g. movies and their genres, people and their professions, stores and their categories, restaurants and their cuisines.

Given this setting, we are interested in the full joint probability distribution $P(C) = P(c_0, c_1 \ldots c_{|C|})$ over the set of categories $C$, where the joint probability of any two categories is: $P_{x \in I}(\{< x, c_1 />, < x, c_2 >\} \subset R'_I)$, generalizing to $N < |C|$ as $P_{x \in I}(\{< x, c_1 >, \ldots, < x, c_N >\} \subset R'_I)$. In other words, the joint probability of some set of categories is the fraction of articles that have all those categories as labels in $R'_I$, the transitive closure of $R_I$. Having rapid access to this joint probability distribution is useful for a lot of tasks, such as recommending or ranking choices. For example, librarians and authors manually label PubMed articles with MeSH, given a set of hand-assigned labels, a user interface could make use of a ranked list of other categories that are likely to be assigned given the initial set.

Computing the full joint distribution requires computing $2^{|C|}$ queries, where each query requires one full iteration over the entire instance dataset in order to count co-occurrences of category labels; with roughly 30k categories in MeSH, the evaluation on the full set of possible queries is intractable. Instead we seek to exploit box model training to generalize from a subset of the total queries to the full joint distribution.

A significant challenge in sampling from the full distribution is that, for any given $N < |C|$, the distribution of $N$-joint probabilities is exponential, with a density function whose rate parameter $\lambda \propto k^N$ (for our large dataset, $k \sim 60$). A few examples of probability density functions for that dataset is shown in Fig. 1. Sampling from this space yields either unrealistic bias or few positive examples.

Our central research question is

**Research Question 1** How well do box embeddings approximate the distribution of all possible joint probabilities, without enumerating them?
3. Box Lattice Embeddings

Box lattice embeddings (box embeddings or boxes) represent each category in $\mathcal{C}$ as an n-dimensional box, and each instance in $\mathcal{I}$ as a fixed sized box, such that the sum of all the instance box volumes is 1.0. The category boxes can change in size, and both instance and category boxes can move in the space. The objective is to draw the boxes in this space such that all instance boxes are inside the category boxes they belong in, following $\mathcal{R}_{I}'$. In other words, represent every edge $\langle i, c \rangle \in \mathcal{R}_{I}'$ as the containment of the box for $i$ in the box for $c$.

Formally, boxes associate each category with 2 vectors in $[0, 1]$, $(x_m, x_M)$, the minimum and maximum value of the box at each dimension [Vilnis et al., 2018]. At the present time, we are not investigating dimensions that have any latent meaning, thus the number of dimensions is just a hyperparameter, with the intuition that larger number of dimensions may be needed to satisfy more complex constraints. For numerical reasons these vectors are stored as a minimum and a positive offset plus an $\epsilon$ term to prevent boxes from becoming too small. This representation of boxes in Cartesian space can define a partial ordering by containment of boxes, and a lattice structure as:

$$x \wedge y = \perp \text{ if } (x, y) \notin \mathcal{R}_{I}' \cup \mathcal{R}_{SC}', \text{ (i.e. } x \text{ and } y \text{ are disjoint), else}$$

$$x \wedge y = \prod_i [\max(x_{m,i}, y_{m,i}), \min(x_{M,i}, y_{M,i})]$$

$$x \vee y = \prod_i [\min(x_{m,i}, y_{m,i}), \max(x_{M,i}, y_{M,i})]$$

if $A = (x_m, x_M)$ then $p(A) = \prod_i^n (x_{M,i} - x_{m,i})$

and the hard box objective function follows as

$$\Phi_H = \sum -\log(p(a \vee b) - p(a) - p(b))$$

In other words, maximize the overlap of boxes with an edge in $\mathcal{R}_{I}' \cup \mathcal{R}_{SC}'$. This objective, while intuitive, creates un-learnable gradients due to the hard edges, and was updated in [Li et al., 2019] to soften the edges of boxes using the softplus function, the antiderivative
of the logistic sigmoid:

\[
\text{soft}(x) = \log(1 + \exp(x))
\]

\[
m_{\text{soft}}^{(i)}(x) = \frac{\text{soft}(x^{\rho^{-1}})}{\text{soft}(\rho^{-1})}
\]

\[
p(x) = \prod_{i} m_{\text{soft}}^{(i)}(x_{M,i} - x_{m,i})
\]

\[
p(x, y) = \prod_{i} m_{\text{soft}}^{(i)}(x_{M,i} \wedge y_{M,i} - x_{m,i} \lor y_{m,i})
\]

In other words, as above, the marginal probability of category \( x \) is the volume of its box \( x \), and the joint probability of two categories is the volume of their intersecting box. One of the desirable properties of boxes is that they are \textit{closed under intersection}, and it should be easy to see how the joint probability calculation above generalizes to an arbitrary sized joint probability.

The temperature parameter \( \rho \) at the zero limit turns these soft boxes into hard boxes and we retain the uniform probability measure of hard boxes, but the authors note that, like a Gaussian model, higher values of \( \rho \) have desirable properties for training, in particular gradient smoothness. Since boxes provide the ability to quickly retrieve joint probabilities and marginals they can be used easily to calculate conditional probabilities as well. Practically speaking, our experiments measure how good an approximation of \( p(x, y) \) or \( p(x|y) \) can be obtained in various training regimens with variations on a softbox loss function using KL divergence between \( P(x, y) \) and \( P_T(x, y) \), the true observed joint probability distribution, or \( P(x|y) \) and \( P_T(x|y) \).

4. Datasets

MeSH, the NLM \textit{Medical Subject Headings}, \cite{NLM2018a} is a taxonomy of subject headings for categorizing medical writing. It is organized in a traditional library subject classification style, with the slight difference that subject headings in the taxonomy can have multiple parents – it is a DAG not a tree. Pubmed \cite{NLM2018b} is a very large collection of metadata for over 30 million medical journal articles, each with human labeled subject categories from MeSH. The medical journal community of authors, editors, publishers and the NLM’s librarians, work together to keep the process scalable and of reasonable quality. That’s not to say there aren’t errors, but no taxonomy is free of errors.

MeSH is organized into 16 top level categories, such as \textbf{A}: Anatomy, \textbf{B}: Organisms, \textbf{C}: Diseases, which themselves cannot be the subjects of articles. The taxonomy is on average 8 levels deep, with a generalization or broader-term relationship from child to parent nodes, e.g. \langle RespSystem, Anatomy \rangle, \langle Larynx, RespSystem \rangle, \langle Asthma, Diseases \rangle.

The complete MeSH 2018 contains roughly 28K category labels, and PubMed 2018 contains roughly 30M articles. For our experiments we generated two subsets, \textit{small} and \textit{large}. The \textit{large} dataset includes the largest 3656 of the 6600 categories in the \textbf{A}: Anatomy and \textbf{C}: Diseases taxonomy, and 436K articles from PubMed that have MeSH labels in that
set. The small dataset is a subset of large, containing 151 categories and 2500 articles\(^1\).

More detail about these sets is shown in Tab. 1.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>small</td>
<td>151</td>
<td>2500</td>
<td>35K</td>
</tr>
<tr>
<td>large</td>
<td>3656</td>
<td>436K</td>
<td>9M</td>
</tr>
</tbody>
</table>

Table 1: Properties of our experimental datasets.

The MeSH Anatomy hierarchy mostly follows a mereological generalization (sub-parts to parts), e.g. \(\langle\text{Bronchi, Lung}\rangle\), while diseases follow a slightly more causal generalization, e.g. \(\langle\text{Pneumonia, BacterialInfection}\rangle\). This kind of semantic promiscuity is extremely common in taxonomies [Guarino and Welty, 2009], and causes an imprecision that begs for an approach with soft, continuous-valued constraints, as opposed to the traditional discrete reading of the taxonomic relationship (all members of the subcategory are members of the super-category). It was this observation that led us to taxonomy embeddings.

Articles listed in PubMed can have any number of subject headings, on average 8-10, and it is fully expected by the published methodology that assigning a particular subject heading to an article also assigns the MeSH parents and ancestors – they expect the transitive closure to hold.

5. Instance vs Category Training

Embeddings are typically trained on pairwise category relations, and use some form of pairwise distance metric as an energy function. This is a natural choice for most problems as many common distance metrics (cosine similarity, inner products, KL divergence) are only defined over two elements, and it avoids the combinatorial problem of enumerating all higher order combinations.

We investigate two relations present in the PubMed/MeSH data for training:

- **inst-cat**: Instance to category edges, in which a box for each instance must be contained within the category boxes for all its edges in \(R'_I\), such that if \(\langle i, c \rangle \in R'_I\), then \(\frac{p(c,i)}{p(i)} \sim 1\).

- **cat-cat**: Conditional probabilities between two categories, in which the boxes for each pair of categories \(x, y\) must overlap such that \(\frac{p(x,y)}{p(y)} \sim p(x|y)\).

In each case the objective cost function is the KL-divergence between the target value (1 in the case of instances, and \(p(x|y)\) in the case of conditionals), between the box model and the training true probability.

The intuition of inst-cat training is the observation that if each instance box is contained within all its categories, those categories should ideally overlap on only the instances they have in common; the optimal drawing of the boxes would accomplish precisely what a joint probability calculation is. Each instance is represented by a fix-sized box with volume \(|I|^{-1}\). In the model we accomplish this by restricting the positive offset vector of instance

---

1. Although we cannot share MeSH and PubMed directly, they are available for free with license terms and we will share these precise lists of identifiers in our datasets upon publication.
boxes from being trained, so instance boxes have only one trainable parameter vector. We quickly found that this intuition requires that instance boxes not be allowed to overlap, otherwise the category box intersections may not reflect the number of instances within them. We accomplish this with a special loss for instances that overlap, adding that to the KL-divergence.

The problem with inst-cat training is the number of parameters, which while linear in $|I|$, is quite large. PubMed contains 30M articles and MeSH 30K categories, and it is a long standing assumption of knowledge-bases that $|I| \gg |C|$. Training that many parameters means roughly that many passes through the training set, implying the computational cost of inst-cat training is polynomial on $|I|$.

We developed cat-cat training in reaction to the long experimental runtimes of inst-cat training, initially just as a proxy for training until we had other aspects of boxes working. This includes a period of time before adopting soft-boxes, when the hinged gradients of hard-boxes proved a significant challenge to model convergence. Calculating the conditionals as training data requires only two passes through $R'_I$ data for $N = 1$ and $N = 2$, ie the marginals and two-way joint probabilities. The model then has only $|C| (\times 2)$ parameter vectors with a similar reduction in training time.

This clear savings leads us to:

**Research Question 2** Can cat-cat training provide a reasonable substitute to estimating the full joint probability distribution as compared to inst-cat training?

Each training method was accompanied by its own negative edge generation method. In the case of inst-cat, we defined a hyper-parameter $\gamma = 0.2$ that controlled the rate of negative sampling. We generate the complement of $R'_I, R_I$, and for each edge $\langle i, c \rangle \in R'_I$, we sample $\gamma$ of the edges from the set $\langle i, x \rangle \in R_I$. For the small dataset, there were roughly 342K negative training examples, for large there were 30M.

For cat-cat training, we simply calculate the conditional probability for every pair of categories, including those that are zero. As the number of categories grows, we can again make use of the $\gamma$ parameter to control sub-sampling, but this was not needed on our datasets.

### 6. Evaluation

We are interested in the ability of these models to faithfully reproduce the full joint probability distribution $P(C) = P(c_0, c_1 \ldots c_{|C|})$ over the set of categories $C$. Since the full joint distribution is too expensive to compute, we sample it to create our evaluation set. We generate a set of queries $Q$, where a query $q \in Q$ specifies a sub-set of categories and their labels, $q = \{(c_i, v_i) : c_i \in C, v_i \in \{\text{true, false}\}\}$. Each query requires computing a joint probability over the specified categories, where the size of the query $N$ is the order of the join.

For each query size $N$, we generate 1000 samples, choosing the $\{\text{true, false}\}$ labels with 50-50 probability for each category. For our dataset the instance to category relations are sparse, and as demonstrated in Fig. 1, larger sized queries on the distribution are extremely likely to have zero probability. In Table 2 we show the fraction of non-zero probability
queries $F_{p>0}$ for both the small and large MeSH datasets, and so to avoid sampling only zero probability queries we limit the size of the queries to $N \leq 9$. Computing the ground truth value of these queries requires counting the number of instances which have the same category labels as the query value.

<table>
<thead>
<tr>
<th>$N$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{p&gt;0}$ small MeSH</td>
<td>0.937</td>
<td>0.8</td>
<td>0.575</td>
<td>0.403</td>
<td>0.285</td>
<td>0.162</td>
<td>0.092</td>
<td>0.07</td>
<td>0.052</td>
</tr>
<tr>
<td>$F_{p&gt;0}$ large MeSH</td>
<td>0.983</td>
<td>0.82</td>
<td>0.616</td>
<td>0.418</td>
<td>0.308</td>
<td>0.196</td>
<td>0.134</td>
<td>0.095</td>
<td>0.048</td>
</tr>
</tbody>
</table>

Table 2: Fraction of non-zero probability $F_{p>0}$ N-sized queries with increasing $N$.

The box embeddings are used to predict the result of query by taking intersections of relevant category boxes. The simplest queries to compute are those where all specified category labels are true as all this requires is computing the intersection of all the corresponding true category boxes. For queries with false label values, we do not directly compute probability of a false category label box $P(c = \text{false})$, but rather the compliment $1 - P(c = \text{true})$.

For mixed label queries we therefore need to compute the intersection of the true category label boxes and subtract the union of the false category label boxes. The inclusion-exclusion principle allows us to express the size of the union of a set of elements $|\bigcup_{i}^{n} c_{i}|$ in terms of the element intersections (and avoid double counting) so that:

$$|\bigcup_{i}^{n} c_{i}| = \sum_{k=1}^{n} (-1)^{k+1} \sum_{1 \leq i_{1} \leq \cdots \leq i_{k} \leq n} |c_{i_{1}} \cap \cdots \cap c_{i_{k}}|$$

And the predicted result of any query can be computed by taking

$$P(\bigcap_{v_{i}=\text{true}} c_{i}) - P(\bigcup_{c_{j}:v_{i}=\text{false}} c_{j})$$

To evaluate the quality of the box model predictions, we split up the ground truth into positive examples (where the query probability is greater than zero) and negative examples (where the query probability is zero). For both positive and negative examples we look at the mean absolute error (MAE), and for the positive examples only we compute the spearman rank correlation, and average percent difference. For both instance-category, and category-category training we chose the best performing model to evaluate. The results on the small MeSH dataset and the large MeSH dataset are reported in 3 and 4 respectively.

7. Discussion and Conclusion

The results show that there is no clear winner. While inst-cat training does a little better across the board on the small dataset, on the large dataset, the story is more balanced. Rank correlation ($\rho$) shows inst-cat performing far worse on large, and for MAE$^+$, with the exception of the marginal ($N = 1$), cat-cat has lower error. Roughly half these differences are not significant (error bars to follow in final version), giving cat-cat training a slight, but not definitive, edge.

It is somewhat surprising that cat-cat training works so well, since training on conditional probabilities only encodes the 2-joint probability distribution, and it is not hard to
think of scenarios where the relationship between three or more categories is not accurately represented by their respective pairwise conditional probabilities, and inst-cat training data should be able to completely represent these cases. As a result, we expected the performance for cat-cat training to degrade as $N$ increased, but this was not the case. One possible explanation is that the high parameter count in the large inst-cat model was too complicated to solve, or that we did not allow the training to proceed long enough. Our experiments ran 1500 epochs with a training set size of 9M (+negs) for inst-cat, compared to 500 epochs on a training size of 716K (+negs) for cat-cat.

For RQ 2, the conclusion so far is clear: the efficiency improvements of cat-cat training are significant, with a $300\times$ decrease in the number of training operations, while yielding equivalent or better performance on the $N$-joint probability problem.
For RQ 1, box embeddings have provided a very good approximation of the $N$–joint probabilities up to $N = 9$, with $\text{MAE}^{+}$ numbers in the $10^{-4}$ range. We see surprisingly little evidence that the error would increase, there is no trend as $N$ increases, which is especially notable for cat-cat training.

Acknowledgments

Omitted for blind review.

References


