Beta Embeddings for Multi-Hop Logical Reasoning in Knowledge Graphs

Introduction:

Reasoning in knowledge graph (KG) is a fundamental problem in artificial intelligence. In essence, it involves answering first-order logic (FOL) queries over KGs using operators: existential quantification, conjunction, disjunction and negation. Present methods are limited to only a subset of first-order logic (FOL) operators. In addition, the negation operator is not supported.

The challenges include the scale of the KG and also the incompleteness. In this paper, they propose Beta Embedding (BETAE) the idea is to model both the entities and queries by probabilistic distributions with bounded support. It is able to handle arbitrary first-order logic queries in an efficient and scalable manner.

Related work:

Previous methods aim to address the challenges by using embeddings and this way implicitly impute the missing edges. There are works on KG embeddings, (2) assign a learnable vector for each entity and relation with various geometric intuitions and neural architectures which mainly focus on link prediction and hard to generalize to multi-hop reasoning. Also, there are works model the uncertainty using order embeddings, distributions and quantum logic. But it's hard to go beyond modeling the inclusion and entailment between a pair of concepts in KGs.

Another line of related work is multi-hop reasoning on KGs. This includes (1) answering queries and (2) use multi-hop rules or paths to improve the link prediction performance. But (1) is limited to subset of FOL and (2) pre-define or achieve these rules in an online fashion which require a modeling of all the intermediate entities on the path and not scalable.

For the two baselines compared later in experiments:
GQEs: The key idea is to learn how to embed any conjunctive graph query into a low-dimensional space. This is achieved by representing logical query operations as geometric operators that are jointly optimized on a low-dimensional embedding space along with a set of node embeddings.

Q2B: The idea is to represent queries as boxes instead of points. Thus the intersection will be straightforward. It can only work on EPFO queries but not with negation. Since it embeds queries as closed regions but negation of a closed region does not result in closed region.

Solution:

the desirable properties of the embedding include: (1) Naturally model uncertainty, (2) We can design logical/set operators (conjunction/intersection, negation/complement that are closed).
In this paper, they embed entities and queries into the same space using probabilistic embeddings with bounded support. Especially, they use beta distribution.
(PDF): $p(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathbf{B}(\alpha,\beta)}$, where $x \in [0,1]$ and $\mathbf{B}(\cdot)$ denotes the beta function.
A beta distribution can be uniquelt represented by its 2 parameters.

In order to answer a query using the computation graph, we need probabilistic logical operators for the Beta embedding. They include projection P, intersection I and negation N as logical operators. For P, they build a MLP to learn the transformation: $\mathbf{S}' = \mathtt{MLP}_r(\mathbf{S})$

And for I, they take the weighted product of the PDFs of the input Beta embeddings:

$$p_{\mathbf{S}_{\mathtt{Inter}}} = \frac{1}{Z}\prod p_{\mathbf{S_1}}^{w_1} \cdots p_{\mathbf{S_n}}^{w_n},$$

They use the attention mechanism and learn the weights through a MLP_att:

$$w_i = \frac{\exp(\mathtt{MLP}_{\mathtt{Att}}(\mathbf{S_i}))}{\sum_j \exp(\mathtt{MLP}_{\mathtt{Att}}(\mathbf{S_j}))}$$

For N, the function is to take Beta embedding S as input and produces an embedding of the complement N(S) as a result. They simply:

- Method: take reciprocal of α, β

$$\mathcal{N}([(\alpha, \beta)]) = [(\tfrac{1}{\alpha}, \tfrac{1}{\beta})]$$

For training, they define the distance between entity v and the query q as the sum of KL divergence

$$\mathtt{Dist}(v; q) = \sum_{i=1}^{n} \mathtt{KL}(p_{\mathbf{v},\mathbf{i}}; p_{\mathbf{q},\mathbf{i}}),$$

Our objective is to minimize the distance between the Beta embedding of a query and its answers while maximizing the distance between the Beta embedding of the query and other random entities via negative sampling which defined below:

$$L = -\log\sigma\left(\gamma - \mathtt{Dist}(v; q)\right) - \sum_{j=1}^{k}\frac{1}{k}\log\sigma\left(\mathtt{Dist}(v'_j; q) - \gamma\right)$$

Experimental results:

They evaluate BETAE on multi-hop reasoning over standard KG benchmark datasets (FB15K, FB15k-237 and NELL995). Two state of the art baselines for answering complex logical queries on KGs are Q2B and GQE. They first compare the methods on EPFO queries with an average 9.4%, 5% and 7.4% improvement on the three datasets. BETAE can also works on both DNF and DM but the baselines can only work on DNF. BETAE can also work on quires with negation. BETAE also achieves up to 77% better correlation than Q2B and can naturally model queries without answers.

Conclusions:

BETAE could handle arbitrary FOL queries on KGs. It significantly outperforms previous state-of-the-art, which can only handle a subset of FOL, in answering arbitrary logical queries as well as modeling the uncertainty.

Pros:

1. Probabilistic modeling can effectively capture the uncertainty of the queries.
2. They design neural logical operators that operate over these Beta distributions and support full first order logic.

3 The neural modeling of ‘and’ and negation naturally corresponds to the real operations and captures several properties of first-order logic.
4 Using the De Morgan’s laws, disjunction V can be approximated with ‘and’ and negation, allowing BETAE to handle a complete set pf FOL operators and thus supporting arbitrary FOL queries.

Cons:

It might make undesirable predictions in a completely random KG, or a KG manipulated by adversarial and malicious attacks.