Learnable Non Euclidean Background Geometry for Embeddings

Bhavya Bhatt [*1]

[1] *Indian Institute of Technology Mandi, Mandi 175005, India*

July 29, 2019

Abstract

This report contains my work that I did in my summer research internship at Siemens Technology and Services Pvt. Ltd., Bengaluru during the period of June 2019-August 2019. These notes are compilation of the current formalism used in embedding learning for knowledge graph and some of my original ideas (heavily influenced with non-euclidean embeddings[1]) which can further be extended for experimentation in code. First section contains small introduction of the mathematical background of the problem of knowledge graph embedding learning and then few more exotic ideas are discussed in the succesive sections. The embedding learning method in machine learning have always targeted to learn relations and entity representations which can be projected into spacially analogous patterns using dimensional reduction techniques like PCA or manifold learning[2](these spacial patterns totally depends on the supervised task which we are using to learn the embeddings from, in our context it would become clear in a moment).

---

[*]Bhavyabhatt17@gmail.com

[1]see references

[2]one famous example is Word2Vec representations in which projected low dimensional manifold shows us that similar context words are closer to each other and dissimilar words are far apart

Contents

1 Introduction

A knowledge graph represents a DAG structure to model facts (or relations) between entities within some particular domain. Let's take an example of relations in a family, here family represents informally a domain, entities are the members of the family and relations represent specific blood relations like $isfather()$, $ismother()$ etc. We will only model two argument relations so for example Let s be son and f be father so according to the above defination we have

$$isFather(s, f) = 1$$
$$isMother(s, f) = 0$$

One observation worth noting is the asymmetry in the relation which states that if we exchange the arguments, the value of the relation no longer remains same (as son is not father of his father) so

$$isFather(f, s) = 0$$

. Now the problem is to find new fact or predict new links between the entities based on the given knowledge graph. For example if we have been given

$$isFather(s, f) = 1$$
$$isWife(f, m) = 1$$

then based on these facts our probabilistic model should be able to predict that

$$isMother(s, m) = 1$$

. Typically in real life application this knowledge graph can be exponentially large and can contain millions of such facts. Now to model the problem we first need vectorized representations of the entities and relations so as to make further mathematical modelling easier. Now these entities and relations should be composed in such a way (using some mathematical operations) that they encapsulate all the knowledge that is represented by the particular triplet[3]. Now people have used many such compostional operations to model these links and we will discuss few popular ones which claim to provide state-of-the-art results on testing dataset. Before that let us first mathematically formalize the problem which we do in the next section.

1.1 Mathematical Preliminary

We have a subject entity s and object entity o such that $s, o \in \Lambda$ where Λ is the set of all entities in the domain and similarly we have set of all relations P for which there exist a r_p such that $r_p \subseteq \Lambda \times \Lambda$ with a mapping function $\phi : \Lambda \times \Lambda \mapsto \{0, 1\}$ to a real value 1 or 0 depending on whether the fact expressed by the relation and entities (passed as arguments) is true or false respectively. We also denote number of entities in Λ with n_e and number of

---

[3]the set of two entity and a relation is called a triplet

relations in P with n_r. We can then formally write our probabilistic model for all triplets $x_i = \{p, s, o\}_i \in P \times \Lambda \times \Lambda$ and labels $y_i \in \{0, 2\}$ with the dataset $D = \{x_i, y_i\}_{i=1}^m$ as follows

$$P(Y_{s,o,p} = 1|\Theta) = \sigma(\eta_{s,o,p}) = \sigma(r_p^\mathsf{T} e_s \circ e_o) \tag{1}$$

where $\Theta = \{e_i\}_{i=1}^{n_e} \cup \{r_p\}_{j=1}^{n_r}$ is set of all parameters and r_p, e_s, e_o are embeddings of relation, subject entity and object entity respectively such that $r_p, e_s, e_o \in \mathbb{R}^K$ where K is number of dimensions of the embedding space and σ is the sigmoid function to scale values between 0 to 1. The operator $\circ$ is the composition operator which should capture the meaning of how the entities combine with the relation to produce the fact as true or false. This operator is an attempt to model the interaction between a pair of entities and a relation in its abstract form and thus is very crucial to make sense of our model. We now discuss different choices of these compositional operator and associated model, each one tries to model different type of interaction[4]. One possible loss function for this supervised learning task[5] is standard logistic loss with L_2 regularizer as follows

$$L(\Theta) = \sum_{i=1}^m \log(1 + \exp\{-y_i \sigma(\eta_{s,o,r})_i\}) + \lambda\|\Theta\|^2 \tag{2}$$

2 Different Models

2.1 Euclidean Tensor Product composition: Modelling symmetric relations

This model assumes a very simple yet computationally expensive compositional operator $\otimes$ which is a tensor product between two vectors and so our above model becomes as follows

$$P(Y_{s,o,p} = 1|\Theta) = \sigma(\eta_{s,o,p}) = \sigma(e_r^\mathsf{T} e_s \otimes e_o) \tag{3}$$

where $r_p \in \mathbb{R}^{K^2}$ and $e_s, e_o \in \mathbb{R}^K$. This model assumes that all the components of the embedded representations (both relation as well as entity) contribute in the output. This model is computationally expensive and many implementations assume a diagonal matrix for r_p and thus this can only model symmetric relations. To overcome this we need some non-linear operation between the entity embeddings to be able to model anti-symmetric relations and at the same time having less total parameters to obey complexity bounds. Hence we move to the next model which tries to achieve this non-linear composition through circular correlation[6].

---

[4]example of a interaction is symmetric relations in which the value of the fact is unaffected with the entity exchange in the argument

[5]this is the supervised learning task in our context which we were talking about in the abstract

[6]more on this can be found in the related reference

2.2 Holographic Embeddings

In this model the composition operator is a so called holographic[7] circular correlation which is defined as follows

$$(a \star b)_k = \sum_{i=0}^{K-1} a_i b_{k+i \ mod \ K} \tag{4}$$

and our model looks as follows

$$P(Y_{s,o,p} = 1|\Theta) = \sigma(\eta_{s,o,p}) = \sigma(e_r^\mathsf{T} e_s \star e_o) \tag{5}$$

where $r_p, e_s, e_o \in \mathbb{R}^K$.

2.3 Complex Embeddings and Hermitian Composition: Modelling anti-symmetric relations

One another way to model anti-symmetric relations to use hermitian product for complex embeddings. So in this model all embeddings are in complex space and the operator used is mean of element-wise product between these complex embeddings of relation and entities with a consideration of two representations for each entity one is $e_o = e_i$ when it acts as object and $e_s = \bar{e}_i$ when it acts as subject where $\bar{e} = (e^\star)^\mathsf{T}$ represents dagger of the complex vector e. The model is as follows

$$\eta_{s,o,p} = Re(< r_p e_s \bar{e}_o >) = Re(\sum_{k=1}^{K} r_{pk} e_{sk} e_{ok}^-) \tag{6}$$

where $r_p, e_s, e_o \in \mathbb{C}^K$. It is apparent that because of the dagger operation the composition is no more symmetric and thus this type of complex embedding is useful in modelling asymmetric or anti-symmetric relations which was not possible with diagonal tensor product composition.

2.4 Complex Embeddings with simple constraints

These are simple constraints on loss functions in case of complex embeddings and details can be refered from the original paper by Ding et al-2018.

3 Learnable Non-Euclidean Embeddings

This section discusses some of the original ideas which the author seems to have after learning different approaches that exist in the literature and attempts to give a unified view on the non-euclidean embedding methods some of which seems to work for specific background geometry (for example Hyperbolic geometry[8]). Some assumptions for this section is

---

[7]term taken from assosiative memory

[8]see the reference for more details on hyperbolic geometry

that embeddings exist in tangent space of a metric manifold whose intrinsic geometry is completely determined by the metric tensor[9]. Now we will briefly discuss about the differential geometric perspective of the infinitesimal distance in any arbitary geometry as follows

---

[9]connections on the manifold are metric compatible - but we can also experiment with some other connections which results in torsion terms in our expressions and explore what results it give on the benchmark dataset