Training Relation Embeddings under Logical Constraints

Pushpendre Rastogi[1] pushpendre@jhu.edu
Adam Poliak[1] azpoliak@cs.jhu.edu
Benjamin Van Durme[1,2] vandurme@cs.jhu.edu

[1]Center for Language and Speech Processing
[2]Human Language Technology Center of Excellence
Johns Hopkins University

Abstract

We present ways of incorporating logical rules into the construction of embedding based Knowledge Base Completion (KBC) systems. Enforcing "logical consistency" in the predictions of a KBC system guarantees that the predictions comply with logical rules such as symmetry, implication and generalized transitivity. Our method encodes logical rules about entities and relations as convex constraints in the embedding space to enforce the condition that the score of a logically entailed fact must never be less than the minimum score of an antecedent fact. Such constraints provide a weak guarantee that the predictions made by our KBC model will match the output of a logical knowledge base for many types of logical inferences. We validate our method via experiments on a knowledge graph derived from WordNet.

1 Introduction

A number of state of the art methods for Knowledge Base Completion (KBC) utilize a representation learning framework and learn distributed vector representations, i.e. *embeddings*, of the entities and relations in a Knowledge Base (KB). Although such models make correct predictions on a sizable portion of the data, they cannot guarantee to follow logical rules and to make inferences consistent with those rules. For example, there is no way to guarantee in existing KBC methods that if an embeddings based KB predicts the fact that *Alice murdered Bob* `(Murdered,(Alice,Bob))` then it will also predict that *Alice Killed Bob*, even though it is very simple to enforce this in a traditional logical inference system by specifying the rule that `Murdered` implies `Killed`.

In this paper we present a novel method for directly encoding logical rules via convex constraints on the embeddings. Such methods for directly "shaping" the feasible subspaces of embeddings based on logical properties of relations have not been deeply explored before, and we will show through our experiments that such a method can improve the performance of an existing KBC system.

2 Method

Let a knowledge base be defined as a tuple $(\mathcal{F},\mathcal{L})$, with $\mathcal{F}$ a set of statements, and a set of first order logic rules $\mathcal{L}$. Every element $f\in\mathcal{F}$ is itself a nested tuple $(r,(e,e'))$ which states that the entities e and e' are connected via the relation r. Let $\mathcal{E}$ and $\mathcal{R}$ be the set of all entities and relations respectively. Let $\mathcal{T}$ be the set of all entity tuples that appear in $\mathcal{F}$, and let $\mathcal{U}$ denote the universe of all possible facts, i.e. $\mathcal{T}=\{t\,|\,(r,t)\in\mathcal{F}\}$, and $\mathcal{U}=\{(r,(e,e'))\,|\;r\in\mathcal{R},\,e,e'\in\mathcal{E}\}$. Note that $\mathrm{T}\leq\mathrm{F}$.[1] Finally, $\mathcal{F}^c=\mathcal{U}\setminus\mathcal{F}$ is the set of unknown facts. The goal of a KBC system is to rank the elements of $\mathcal{F}^c$ so that facts that are correct receive a smaller rank than incorrect facts.

Embedding Model: We assume that every relation $r\in\mathcal{R}$ and entity $e\in\mathcal{E}$ can be represented using real valued vectors $\mathbf{r}\in\mathbb{R}^d$ and $\mathbf{e}\in\mathbb{R}^{\tilde{d}}$; d and $\tilde{d}$ may have different values. The vector representation of each tuple t is computed from its constituent entities via a composition function $c:\mathbb{R}^{\tilde{d}}\times\mathbb{R}^{\tilde{d}}\to\mathbb{R}^d$, i.e. $\mathbf{t}=c(\mathbf{e},\mathbf{e}')$. For example c may denote vector concatenation, in which case

In: L. Dietz, C. Xiong, E. Meij (eds.): Proceedings of the First Workshop on Knowledge Graphs and Semantics for Text Retrieval and Analysis (KG4IR), Tokyo, Japan, 11-Aug-2017, published at http://ceur-ws.org

[1]Per standard convention we denote the size of a set using the corresponding roman symbol. E.g. E is the size of $\mathcal{E}$.

$\mathbf{t}=[\mathbf{e}^T,\mathbf{e'}^T]^T$. We will use the semicolon symbol ; as an infix operator to denote vector concatenation, i.e. $(\mathbf{x};\mathbf{y})=[\mathbf{x}^T,\mathbf{y}^T]^T$. Finally, $\mathbf{x}\geq\mathbf{y}$ denotes that the vector $\mathbf{x}$ is elementwise larger than $\mathbf{y}$ and $B(\mathbf{x},r)$ denotes the L_2 ball centered at $\mathbf{x}$ with radius r.

Score Function: A majority of the existing work on embedding based KBC measures the *correctness* of a fact via a scoring function, $score: \mathcal{R}\times\mathcal{E}\times\mathcal{E}\rightarrow\mathbb{R}$, with the property that when $score(f)>score(f')$, fact f is more likely to be correct than f'. The two major classes of score functions are:

$$score(f)=\langle\mathbf{r},\mathbf{t}\rangle \tag{1}$$

$$score(f)=-||\mathbf{r}-\mathbf{t}||^2 \tag{2}$$

In Equations (1–2), $\mathbf{r}$ and $\mathbf{t}$ are vector representations of r and $t=(e,e')$, respectively, that are constituents of $f=(r,(e,e'))$. For brevity, we will omit this expansion from here onwards.

Unconstrained Objectives for Learning Score Function Rendle et al. 2009 proposed the Bayesian Personalized Ranking (BPR) objective as a way of tuning recommendation systems when a user can only observe positive training data, such as correct facts, but the facts that are absent may be either correct or incorrect. In this paper we will focus on the BPR objective since this objective has been used for learning the parameters of a KBC system by various researchers Rendle et al. (2009); Demeester et al. (2016); Riedel et al. (2013). Wang and Cohen 2016 experimentally showed that the BPR objective outperforms other objectives such as Hinge Loss and Log Loss.

BPR posits that the training data is a single joint sample of $\mathrm{U}(\mathrm{U}-1)$ bernoulli random variables $\{b_{ff'}\mid f\in\mathcal{U},f'\in\mathcal{U},f'\neq f\}$. $b_{ff'}$ equals 1 when f is in $\mathcal{F}$ and f' is in $\mathcal{F}^c$ and 0 otherwise. $b_{ff'}$ is parameterized by its probability $p_{ff'}$ and all $b_{ff'}$ are conditionally independent given the probabilities $p_{ff'}$. The probability values must obey the reasonable condition that $p_{ff'}=1-p_{f'f}$. A natural way to satisfy this condition is to parameterize $p_{ff'}$ as $\sigma(score(f)-score(f'))$ where σ is the sigmoid function.[2] The BPR estimator is simply the L_2 regularized MLE estimator of this probabilistic model, with regularization strength α. Table 1 lists instances of the BPR objective that arise with different score functions.

Model	*score*	Minimization Objective (J)
A $\mid t=(\mathbf{e};\mathbf{e}')$ R $\mid t=\mathbf{e}\otimes\mathbf{e}'$	(1)	$-\sum_{(f,\bar{f})\in\mathcal{F}\times\mathcal{F}^c}\log(\sigma(\langle\mathbf{r},\mathbf{t}\rangle-\langle\bar{\mathbf{r}},\bar{\mathbf{t}}\rangle))+\alpha(\sum_{r\in\mathcal{R}}\|\mathbf{r}\|^2+\sum_{t\in\mathcal{E}}\|\mathbf{e}\|^2)$
B $t=(\mathbf{e};\mathbf{e}';\mathbf{e}^T\mathbf{e}')$	(1)	$-\sum_{(f,\bar{f})\in\mathcal{F}\times\mathcal{F}^c}\log(\sigma(\langle\mathbf{r}_1,\mathbf{e}\rangle+\langle\mathbf{r}_2,\mathbf{e}'\rangle+\langle\mathbf{e},\mathbf{e}'\rangle-\langle\bar{\mathbf{r}}_1,\bar{\mathbf{e}}\rangle-\langle\bar{\mathbf{r}}_2,\bar{\mathbf{e}}'\rangle-\langle\bar{\mathbf{e}},\bar{\mathbf{e}}'\rangle))+\alpha(\sum_{r\in\mathcal{R}}\|\mathbf{r}\|^2+\sum_{t\in\mathcal{E}}\|\mathbf{e}\|^2)$
C $\mid t=(\mathbf{e};\mathbf{e}')$ T $\mid t=\mathbf{e}-\mathbf{e}'$	(2)	$-\sum_{(f,\bar{f})\in\mathcal{F}\times\mathcal{F}^c}\log(\sigma(\|\bar{\mathbf{r}}-\bar{\mathbf{t}}\|^2-\|\mathbf{r}-\mathbf{t}\|^2))+\alpha(\sum_{r\in\mathcal{R}}\|\mathbf{r}\|^2+\sum_{t\in\mathcal{E}}\|\mathbf{e}\|^2)$
D $t=(\mathbf{e};\mathbf{e}'\|\mathbf{e}-\mathbf{e}'\|)$	(2)	$-\sum_{(f,\bar{f})\in\mathcal{F}\times\mathcal{F}^c}\log\sigma\begin{pmatrix}\|\bar{\mathbf{r}}_1-\bar{\mathbf{e}}\|^2+\|\bar{\mathbf{r}}_2-\bar{\mathbf{e}}'\|^2+\|\bar{\mathbf{e}}-\bar{\mathbf{e}}'\|^2\\-\|\mathbf{r}_1-\mathbf{e}\|^2-\|\mathbf{r}_2-\mathbf{e}'\|^2-\|\mathbf{e}-\mathbf{e}'\|^2\end{pmatrix}+\alpha\left(\sum_{r\in\mathcal{R}}\|\mathbf{r}\|^2+\sum_{t\in\mathcal{E}}\|\mathbf{e}\|^2\right)$

Table 1: Instances of the BPR objective corresponding to different choices of composition and score functions. For example, if $c(\mathbf{e},\mathbf{e}')=(\mathbf{e};\mathbf{e}')$ and Equation 1 is used as the score function then we need to minimize the function in the first row with respect to $\mathbf{r},\mathbf{e}$. In the first and third row, $\mathbf{r}=(\mathbf{r}_1;\mathbf{r}_2)$, in the second row $\mathbf{r}=(\mathbf{r}_1;\mathbf{r}_2;1)$ and in last row $\mathbf{r}=(\mathbf{r}_1;\mathbf{r}_2;0)$. The symbol $\otimes$ refers to the vector outer product operator that takes two vectors of size $\tilde{d}$ and produces a vector of size $\tilde{d}^2$. Since this scoring function is equivalent to the score function of RESCAL we call the model R. Similarly the scoring function for T is the same as TransE.

Logical Consistency of Embeddings through Constraints Our general scheme for incorporating logical relations into embeddings is to ensure that during the learning of the vector representation of entities and relations, the score of a consequent fact will be greater than the score of any of its antecedents. In other words if $f_1,\ldots,f_{n-1}\implies f_n$ then $score(f_n)\geq\min_{i\in[1,n-1]}(score(f_i))$. If this does not hold true, then it will be possible for our KB to assign a low score to a logically entailed fact even though all of its antecedents have a high score.

We analyze common logical rules found in large scale KBs and for different combinations of a logical rule and scoring function we devise inequalities that the score function should satisfy. We translate those inequalities into constraints that restrict the entity and relation representations in a KB. We use the projected subgradient descent algorithm for learning the parameters of our KBS system. Algorithm 1 shows a specific instance, for Model A and batch size 1, of our parameter learning algorithm with a general set of rules $\mathcal{L}$. We now show how to construct convex constraints from logical rules.

2.1 Constraints for Logical Consistency: Relational Implication

We now present the constraints for guaranteeing that the predictions from embeddings based KBC systems are consistent with logical rules starting with implication rules. An implication rule of the form, $\textsc{RelImp}(r,r')$, specifies that if a fact $f=(r,t)$ is

[2]The sigmoid function, $\sigma(x)=\frac{1}{1+\exp(-x)}$, has the useful properties that $\sigma(x)+\sigma(-x)=1$ and $\frac{d\sigma(x)}{dx}=\sigma(x)\sigma(-x)$.

correct, then (r',t) must also be correct. For example, the rule RELIMP(`HusbandOf, SpouseOf`) enforces that if our KB predicts the fact, `(HusbandOf, (Don, Mel))`, then it will also predict `(SpouseOf, (Don, Mel))`. As explained above we can enforce such a rule by ensuring that *score*$(r',t) \geq$ *score*(r,t) $\forall t \in \mathcal{T}$. [3]

When we use the inner product score function (1) then this inequality can be enforced by ensuring that $\langle \mathbf{r}'-\mathbf{r},\mathbf{t}\rangle \geq 0$ for all $t \in \mathcal{T}$. We constrain $\mathbf{t}$ to lie in a subset of $\mathbb{R}^d$, say $\mathbb{T}$, then the implication rule can be enforced by constraining $\mathbf{r}'-\mathbf{r}$ to lie in the dual cone of $\mathbb{T}$, denoted $\mathbb{T}^*$. A very convenient special case arises when we chose $\mathbb{T}$ to be a "self dual cone" for which $\mathbb{T}=\mathbb{T}^*$. The set of positive real vectors $\mathbb{R}^d_+$ is one example of such a self dual cone. [4]

When we use the L_2 distance score function (2) then the restriction on the score function translates into the following constraints on the vector representations: $||\mathbf{r}-\mathbf{t}||^2-||\mathbf{r}'-\mathbf{t}||^2 \geq 0 \implies \langle \mathbf{r}-\mathbf{r}',\mathbf{r}'+\mathbf{r}-2\mathbf{t}\rangle \geq 0 \implies \langle \mathbf{r}-\mathbf{r}',\mathbf{r}'+\mathbf{r}\rangle/2 \geq h_{\mathbb{T}}(\mathbf{r}-\mathbf{r}')$. Here $h_{\mathbb{T}}(\mathbf{x})$ is the value of the support function of $\mathbb{T}$ at $\mathbf{x}$ which is defined as $h_{\mathbb{T}}(\mathbf{x})=\sup_{\mathbf{t}\in\mathbb{T}}\langle \mathbf{x},\mathbf{t}\rangle$. It is necessary and sufficient for the feasibility of this constraint that the $h_{\mathbb{T}}$ function should be finite for at least one value of $\mathbf{x}=\mathbf{r}-\mathbf{r}'$. Once we have fixed $\mathbf{r}-\mathbf{r}'$ then $\mathbf{r}+\mathbf{r}'$ can be easily chosen from the halfspace $H^-(\mathbf{r}'-\mathbf{r},2h_{\mathbb{T}}(\mathbf{r}-\mathbf{r}'))$. Note that if $h_{\mathbb{T}}$ is difficult to compute then implementing this constraint will also be difficult, therefore we must chose $\mathbb{T}$ wisely.

One example of a good choice of $\mathbb{T}$ is $\mathbb{R}^d_+$. $h_{\mathbb{R}^d_+}(\mathbf{r}-\mathbf{r}')$ is finite and zero iff $\mathbf{r}-\mathbf{r}' \leq 0$ therefore, the value of $\mathbf{r}+\mathbf{r}'$ must lie in the halfspace $\langle \mathbf{r}-\mathbf{r}',\mathbf{r}'+\mathbf{r}\rangle \geq 0$. Unfortunately, the problem of finding $\mathbf{r}$ and $\mathbf{r}'$ vectors that satisfy this constraint is non-convex and it is not possible to project on to this set given a pair of vectors that violate the constraints. We remedy this situation by adding an additional constraint that $\mathbf{r}+\mathbf{r}'$ must also lie in the negative orthant, i.e. $\mathbf{r}+\mathbf{r}' \leq \mathbf{0}$. Table 2 presents all the derived constraints. Unfortunately, since the T model defines $\mathbf{t}=\mathbf{e}-\mathbf{e}'$, therefore it is not possible to set $\mathbb{T}=\mathbb{R}^d_+$. In the case of the T model if we constrain $\mathbf{e}$ to lie in $B(\mathbf{0},\rho)$ then $\mathbf{t}$ must lie in $B(\mathbf{0},2\rho)$ and $h_{\mathbb{T}}(\mathbf{r}-\mathbf{r}')=2\rho(\mathbf{r}-\mathbf{r}') \implies \frac{\langle \mathbf{r}-\mathbf{r}',\mathbf{r}'+\mathbf{r}\rangle}{||\mathbf{r}-\mathbf{r}'||} \geq 4\rho$. One way to make this constraint amenable to efficient projection is to enforce that $\mathbf{r}+\mathbf{r}'=4\rho(\mathbf{r}-\mathbf{r}')$ and $||\mathbf{r}-\mathbf{r}'||^2 \geq 1 \implies ||\mathbf{r}'||^2 \geq |2\rho-.5|$. This constraint becomes trivial if $\rho=0.25$

2.1.1 Reverse Relational Implication and Symmetry

A reverse relational implication rule denoted by REVIMP(r,r') specifies that if $(r,(x,y))$ is correct, then $(r',(y,x))$ is also correct for all $(x,y) \in \mathcal{T}$. This rule can be enforced through the inequality that *score*$(r',y,x) \geq$ *score*(r,x,y). Depending on the model let $\mathbf{r}=(\mathbf{r}_1;\mathbf{r}_2)$ or $(\mathbf{r}_1;\mathbf{r}_2;1 \text{ or } 0)$ as shown in Table 1, and similarly decompose $\mathbf{r}'$. We will omit this detail in later sections. Under models A and B, this inequality translates to the following constraint $\langle \mathbf{y},\mathbf{r}'_1\rangle+\langle \mathbf{x},\mathbf{r}'_2\rangle \geq \langle \mathbf{x},\mathbf{r}_1\rangle+\langle \mathbf{y},\mathbf{r}_2\rangle$ and under models C and D, the necessary constraints are $\langle \mathbf{r}_1-\mathbf{r}'_2,\mathbf{r}_1+\mathbf{r}'_2-2\mathbf{x}\rangle \geq \langle \mathbf{r}'_1-\mathbf{r}_2,\mathbf{r}'_1+\mathbf{r}_2-2\mathbf{y}\rangle$. Stronger versions of these constraints, which are more efficient to enforce, are shown in Table 2.

A symmetry rule denoted as SYMM(r) specifies that if the fact $(r,(e,e'))$ is known to be correct then $(r,(e',e))$ is also correct. We can only comply with this logical rule in an embedding base KB by ensuring that *score*(r,e,e') = *score*(r,e',e). Under all 4 score models this rule can be enforced only by ensuring that $\mathbf{r}_1=\mathbf{r}_2$.

2.1.2 Entailment

A type A entailment logical rule denoted as ENTAIL$_A(r,e,r',e')$ specifies that $(r,(e,x))$ implies $(r',(e',x))$ for all x in $\mathcal{E}$.[5] A Type B entailment rule, ENTAIL$_B(r,e,r',e')$ specifies that $(r,(x,e))$ implies $(r',(x,e'))$. r and r' may denote the same relation. For example, the rule ENTAIL$_B$(`IsA, Man, IsA, Mortal`) can be used to enforce that if `(IsA, (Socrates, Man))` then the KB must also predict that `(IsA, (Socrates, Mortal))`. The final constraints required to implement a type B entailment rule are shown in Table 3.[6]

2.1.3 Property Transitivity

A property transitivity rule denoted PROTRANS(r,r',e',r'',e'') specifies that if $(r,(x,y))$ and $(r',(y,e'))$ are true then $(r'',(x,e''))$ is also true. For example, the rule PROTRANS(`Partner, Convicted, Criminal, Suspected, Criminal`) can be used to incorporate the common sense rule that if an entity is the partner of a convicted criminal then it will be suspected of being a criminal into the embeddings based KB. Note that the score of the hypothesis fact $(r'',(x,e''))$ should be high if the antecedent facts have high

[3] We abuse notation in saying that *score*$(r,(x,y))=$ *score*(r,x,y).

[4] Other self-dual cones, distinct from $\mathbb{R}^d_+$ also exist such as the Lorentz cone $\{x \in \mathbb{R}^d \mid x_d \geq \sqrt{\sum_{i=1}^{d-1} x_i^2}\}$. We refer the reader to Gruber (2007) for more details on the geometry of closed convex cones and their polar and dual sets.

[5] We use the term, entailment, in the sense of entailment of properties. Note that this is different from implication.

[6] Details: To implement a type B entailment rule we need to ensure that *score*$(r',x,e') \geq$ *score*(r,x,e) for all $x \in \mathcal{E}$. Under model A this inequality translates to, $\langle \mathbf{r}'_1-\mathbf{r}_1,\mathbf{x}\rangle \geq \langle \mathbf{r}_2,\mathbf{e}\rangle-\langle \mathbf{r}'_2,\mathbf{e}'\rangle$. Model B requires $\langle \mathbf{r}'_1-\mathbf{r}_1+\mathbf{e}'-\mathbf{e},\mathbf{x}\rangle \geq \langle \mathbf{r}_2,\mathbf{e}\rangle-\langle \mathbf{r}'_2,\mathbf{e}'\rangle$. Model C requires $\langle \mathbf{r}_1-\mathbf{r}'_1,\mathbf{r}_1+\mathbf{r}'_1-2\mathbf{x}\rangle \geq \langle \mathbf{r}'_2-\mathbf{e}'+\mathbf{r}_2-\mathbf{e},\mathbf{r}'_2-\mathbf{e}'-\mathbf{r}_2+\mathbf{e}\rangle$, and finally the constraints over model D's score functions are $\langle \mathbf{r}_1-\mathbf{r}'_1,\mathbf{r}_1+\mathbf{r}'_1\rangle+\langle \mathbf{e}-\mathbf{e}',\mathbf{e}+\mathbf{e}'\rangle-\langle \mathbf{r}'_2-\mathbf{e}'-\mathbf{r}_2+\mathbf{e},\mathbf{r}'_2-\mathbf{e}'+\mathbf{r}_2-\mathbf{e}\rangle \geq 2\langle \mathbf{r}_1-\mathbf{r}'_1-\mathbf{e}-\mathbf{e}',\mathbf{x}\rangle$.

score for any possible entity y. A natural way in which we can incorporate such a rule into score based KBC models is by ensuring that $score(r'',x,e'') \geq \max_{y \in \mathcal{E}} \min(score(r,x,y), score(r',y,e'))$. In order to derive efficient constraints that can enforce this inequality we strengthen the constraint imposed on the score function by replacing the $\min$ function in the lower bound to a convex combination of the scores, i.e. let $\lambda \in (0,1)$, we enforce the inequality that $score(r'',x,e'') \geq \max_{y \in \mathcal{E}} \lambda score(r,x,y) + (1-\lambda) score(r',y,e')$.

Since a convex combination of two values is greater than their minimum, this stronger inequality translates to the following constraint for model A: $\langle \mathbf{e}'', \mathbf{r}_2'' \rangle - (1-\lambda)\langle \mathbf{e}', \mathbf{r}_2' \rangle + \langle \mathbf{x}, \mathbf{r}_1'' - \lambda \mathbf{r}_1 \rangle \geq \langle \mathbf{y}, \lambda \mathbf{r}_2 + (1-\lambda)\mathbf{r}_1' \rangle$. Let $\mathbf{a} = \frac{\mathbf{r}_1'' - \lambda \mathbf{r}_1 + \mathbf{e}''}{\lambda}$, $\mathbf{b} = -\frac{(1-\lambda)(\mathbf{r}_1' + \mathbf{e}') + \lambda \mathbf{r}_1}{\lambda}$, $c = \frac{\langle \mathbf{r}_2'', \mathbf{e}'' \rangle - (1-\lambda)\langle \mathbf{r}_2', \mathbf{e}' \rangle}{\lambda}$, and let $\mathbb{E}$ contain the set $\{\mathbf{e} \mid e \in \mathcal{E}\}$. For Model B, the above inequality on the score function leads to the the constraint: $\forall \mathbf{x}, \mathbf{y} \in \mathbb{E}$, $\langle \mathbf{x}, \mathbf{y} \rangle \leq \langle \mathbf{x}, \mathbf{a} \rangle + \langle \mathbf{y}, \mathbf{b} \rangle + c$. Remember that our goal is to devise a set $\mathbb{E}$, and constraints on relation embeddings such that it is efficient to project onto it and for which the above inequality can be guaranteed. The following proposition – proof ommitted for lack of space – shows how to construct such a set:

Proposition 1. *Let* $\mathbf{x}, \mathbf{y}$ *be members of* $\mathbb{R}_+^d \cap B(\mathbf{a}, ||\mathbf{a}||)$ *and* $\mathbf{a} \geq \mathbf{0}$ *then* $\langle \mathbf{x}, \mathbf{y} \rangle \leq \langle \mathbf{x}+\mathbf{y}, \mathbf{a} \rangle$.

The above proposition shows that if $\mathbf{a} = \mathbf{b}$ and $c \geq 0$ then by setting $\mathbb{E} = \mathbb{R}_+^d \cap B(\mathbf{a}, ||\mathbf{a}||)$ we can satisfy the above constraints.

Alg. 1 Projected SGD for Model A, Batch Size=1

Given: $\mathcal{F}, \mathcal{F}^c, \mathcal{L}$. Hyperparameters: α, η, S.
for each fact $f \in \mathcal{F}$ **do**
  for S steps **do**
    Sample $\bar{f} = (\bar{t}, \bar{r})$ from $\mathcal{F}^c$
    Let $v = \sigma(\langle \bar{\mathbf{r}}, \bar{\mathbf{t}} \rangle - \langle \mathbf{r}, \mathbf{t} \rangle)$
    ▷ Fix $\mathbf{e}$ and optimize J
    $\frac{\partial J^{(f)}}{\partial \mathbf{r}} = -\mathbf{t}v$, $\frac{\partial J^{(f)}}{\partial \bar{\mathbf{r}}} = \bar{\mathbf{t}}v$
    $(\mathbf{r}; \bar{\mathbf{r}}) \leftarrow proj_{\mathcal{L}}\Big((\mathbf{r}; \bar{\mathbf{r}}) - \eta((\frac{\partial J^{(f)}}{\partial \mathbf{r}}; \frac{\partial J^{(f)}}{\partial \bar{\mathbf{r}}}) + 2\alpha(\mathbf{r}; \bar{\mathbf{r}}))\Big)$
    ▷ Fix $\mathbf{r}$ and optimize J
    $\frac{\partial J^{(f)}}{\partial \mathbf{t}} = -\mathbf{r}v$, $\frac{\partial J^{(f)}}{\partial \bar{\mathbf{t}}} = \bar{\mathbf{r}}_1 v$
    $(\mathbf{t}; \bar{\mathbf{t}}) \leftarrow Proj_{\mathcal{L}}\Big((\mathbf{t}; \bar{\mathbf{t}}) - \eta((\frac{\partial J^{(f)}}{\partial \mathbf{t}}; \frac{\partial J^{(f)}}{\partial \bar{\mathbf{t}}}) + 2\alpha(\mathbf{t}; \bar{\mathbf{t}}))\Big)$

Rule	Model	Constraints
RELIMP(r,r')	A, R, B	$\mathbf{r} \leq \mathbf{r}'$
	C, D	$\mathbf{r} \leq \mathbf{r}' \leq -\mathbf{r}$
REVIMP(r,r')	A,B	$\mathbf{r}_2' \geq \mathbf{r}_1, \mathbf{r}_1' \geq \mathbf{r}_2$
	C, D	$\mathbf{r}_1 \leq \mathbf{r}_2' \leq -\mathbf{r}_1, \mathbf{r}_2 \leq \mathbf{r}_1' \leq -\mathbf{r}_2$.
	R	$matrix(\mathbf{r}') \geq matrix(\mathbf{r})$

Table 2: Constraints sufficient for enforcing RELIMP(r,r') and REVIMP(r,r') The constraint $\mathbf{e} \geq \mathbf{0} \forall e \in \mathcal{E}$ applies for all models. *matrix* is the inverse of the operation that converts a matrix to a vector by concatenating its columns. I.e. $matrix(\mathbf{r})$ denotes the matrix form of the vector $\mathbf{r}$.

2.1.4 Type Implication

A type implication rule, denoted as TYPEIMP(r,e,r'), specifies that if the fact $(r,(x,y))$ is correct then $(r',(x,e))$ is also correct $\forall (x,y) \in \mathcal{T}$. In other words, this rule enforces that positional arguments of a relation possess certain properties. For example, the rule TYPEIMP(`Husbandof`, `Man`, `Gender`) can enforce that if our KB predicts the fact that `(Husbandof, (Don, Mel))` then it also predicts that `(Gender, (Don, Man))`.

Under model A the TYPEIMP(r, e, r') rule translates to the following inequality for the parameters $\langle \mathbf{x}, \mathbf{r}_1' \rangle - \langle \mathbf{x}, \mathbf{r}_1 \rangle \geq \langle \mathbf{y}, \mathbf{r}_2 \rangle - \langle \mathbf{e}, \mathbf{r}_2' \rangle \forall (x,y) \in \mathcal{T}$. Let $\mathbf{a} = \mathbf{e} + \mathbf{r}_1' - \mathbf{r}_1$, $\mathbf{b} = -\mathbf{r}_2$ and $c = \langle \mathbf{r}_2', \mathbf{e} \rangle$. Under model B, the restriction on the score function translates to: $\langle \mathbf{x}, \mathbf{y} \rangle \leq \langle \mathbf{a}, \mathbf{x} \rangle + \langle \mathbf{b}, \mathbf{x} \rangle + c$. The analysis for this case again relies on Propostion 1 and the analysis for models C and D is yet out of reach. See Table 3.

Model	Constraints
A	$\mathbf{r}_1' \geq \mathbf{r}_1, \langle \mathbf{r}_2, \mathbf{e} \rangle \leq \langle \mathbf{r}_2', \mathbf{e}' \rangle$
B	$\mathbf{r}_1' \geq \mathbf{r}_1 + \mathbf{e} - \mathbf{e}', \langle \mathbf{r}_2, \mathbf{e} \rangle \leq \langle \mathbf{r}_2', \mathbf{e}' \rangle$
C	$\mathbf{r}_1 \leq \mathbf{r}_1' \leq -\mathbf{r}_1, \mathbf{r}_2 - \mathbf{e} \leq \mathbf{r}_2' - \mathbf{e}',$ $\mathbf{r}_2' + \mathbf{r}_2 \leq \mathbf{e}' + \mathbf{e}$
D	$\mathbf{r}_1 - \mathbf{r}_1' \leq \mathbf{e}' - \mathbf{e}, \mathbf{e} \geq \mathbf{e}', \mathbf{r}_1 \leq \mathbf{r}_1' \leq -\mathbf{r}_1,$ $\mathbf{r}_2 - \mathbf{e} \leq \mathbf{r}_2' - \mathbf{e}', \mathbf{r}_2' + \mathbf{r}_2 \leq \mathbf{e}' + \mathbf{e}$

Table 3: Sufficient constraints for ENTAIL$_B(r,e,r',e')$. The constraint $\mathbf{e} \geq \mathbf{0} \forall e \in \mathcal{E}$ applies for all models.

Rule	Model	Constraints
PROTRANS	A	$\mathbf{r}_1'' \geq \lambda \mathbf{r}_1, \lambda \mathbf{r}_2 + (1-\lambda)\mathbf{r}_1' \leq \mathbf{0}$ $\langle \mathbf{e}'', \mathbf{r}_2'' \rangle \geq (1-\lambda)\langle \mathbf{e}', \mathbf{r}_2' \rangle$
	B	$\mathbf{r}_1'' + \mathbf{e}'' + (1-\lambda)(\mathbf{r}_1' + \mathbf{e}') = 0,$ $\langle \mathbf{r}_2'', \mathbf{e}'' \rangle \geq (1-\lambda)\langle \mathbf{r}_2', \mathbf{e}' \rangle$,and $\mathbf{a} \geq \mathbf{0}, \forall x \in \mathcal{E}, \mathbf{x} \in \mathbb{R}_+^d \cap B(\mathbf{a}, ||\mathbf{a}||)$
TYPEIMP	A	$\mathbf{r}_1' \geq \mathbf{r}_1, \langle \mathbf{e}, \mathbf{r}_2' \rangle \geq 0$ and $\mathbf{r}_2 \leq \mathbf{0}$
	B	$\mathbf{e} + \mathbf{r}_1' = \mathbf{r}_1 - \mathbf{r}_2, \langle \mathbf{r}_2', \mathbf{e} \rangle > 0$,and $-\mathbf{r}_2 \geq \mathbf{0}, \forall x \in \mathcal{E} \mathbf{x} \in \mathbb{R}_+^d \cap B(-\mathbf{r}_2, ||\mathbf{r}_2||)$

Table 4: Constraints for enforcing PROTRANS$(r,\ r',\ e',\ r'',\ e'')$ and TYPEIMP(r,e,r'). $\mathbf{a} = \frac{\mathbf{r}_1'' - \lambda \mathbf{r}_1 + \mathbf{e}''}{\lambda}$

3 Related Work

The problem of enforcing consistency between the predictions made by a machine learning system and a first order logic system, which is what our work attempts to do, has a large history of research but we will only be able to review recent work on learning representations of entities and relations of a knowledge graph and refer the reader to reviews of neural-symbolic systems Garcez et al. (2002); Hammer and Hitzler (2007) for more references.

Grefenstette 2013 presented a novel model for simulating propositional logic with the help of tensors, however their model relied on high-dimensional boolean embeddings of the entities and relations, and it only guaranteed adherence to the RELIMP rule out of the ones presented in this paper. Rocktäschel et al. 2014; Rocktäschel et al. 2015 generalized Grefenstette's work learning embeddings of entities and relations that were real valued and low dimensional and their learning mechanism could accomodate arbitrary first order logic formulae into the parameter learning objective by propositionalizing the formulae. Their method has two drawbacks in comparison to our proposal — 1) The process of propositionalization can be very expensive, especially for rules like PROTRANS and TYPEIMP that quantify over tuples of entities, and 2) Their method of *differentiation through logic* does not guarantee that the learnt embeddings will always be able to predict unseen relations that are logically entailed given the rules and the training data.

Bowman et al. 2015a,b presented a neural network based method for predicting the existence of natural logic relations between two entities. Their approach too had the drawback that it could not guarantee the inference of logically entailed facts. Guo et al. 2015 presented a method based on LLE Roweis and Saul (2000) for incorporating side information in the form of semantic categories of entities but their method is not capable of incorporating the range of logical rules that we can. Demeester et al. 2016 and Vendrov et al. 2016 proposed an approach to constrain the learnt embeddings in a way that is identical to the method prescribed by us in Subsection 2.1. Our work generalizes their approach in two ways — Firstly, we generalize their proposed constraints by using the language of convex geometry, and secondly, we propose constraints for many more logical rules and score functions than either of the two papers. Wang and Cohen 2016 presented a novel method of factorizing the adjacency matrix of a proof graph of a probabilistic logic language to learn embeddings of first order logic formulas. Our method is conceptually simpler than theirs and requires fewer training stages. Finally, Guo et al. 2016 proposed an alternative method for embedding rules and entities based on t-norm fuzzy logics which was very similar to Rocktäschel et al. 2015's approach.

4 Experiment: Logical Deduction and Knowledge Base Completion on WordNet

Our method for training embeddings based KBC systems allows for a very interesting application of solving logical puzzles using an embeddings based KBC system without using an external logical-symbolic subsystem. We perform a controlled experiment where we compare the performance of an embedding based KBC system trained with the constraints versus a system that has been trained without those constraints.

Data Consider the logical deduction problem shown in Table 5. This is a simplified version of a logical puzzle presented in Russell et al. 1995. In this puzzle, `Nono` is a country that possesses a WMD and `Benedict` has traded with `Nono`. The KB has to deduce whether `Benedict` is a criminal based on just two input facts and 3 rules. The total number of facts is $5^2 \times 4 = 100$.

Rules
RELIMP(`TradeWith`, `TransactWith`)
ENTAIL$_B$(`Possess`, `WMD`, `Considered`, `Enemy`)
PROTRANS(`TransactWith`, `Enemy`, `Considered`, `Criminal`, `Considered`)
Facts
(`Possess`, (`Nono`, `WMD`))
(`TradeWith`, (`Benedict`, `Nono`))
Query ?
(`Considered`, (`Benedict`, `Criminal`))

Table 5: A Logical Deduction Problem. Based on the rules and facts a KB should infer that `Benedict` is a `Criminal`.

	Baseline			ELKB		
Model	P@10	MRR	MAP	P@10	MRR	MAP
A	0.00	0.02	0.01	**0.20**†	**0.44**†	**0.83**†
B	0.00	0.03	0.03	**0.17**†	**0.26**†	**0.35**†

Table 6: Table of Results. The baseline of R is equivalent to the RESCAL method.Bold marks that the average performance is higher. † implies that the difference is significant with two tailed p-value ≤ 0.005 as measured by a matched pair t-test.

Evaluation We train two versions of two KBC systems, Models A and B, with batch size$=1$, $\alpha = 0.001$,$\eta = 0.1$,$S = 200$,$d = 50$, and $\tilde{d} = 25$ using Algorithm 1. Both KBs were trained in one pass using the two training facts. The only difference was that the baseline system did not constrain the embeddings to obey logically derived geometric constraints. After training we queried the KBs for the scores of all possible facts. We ranked all the facts based on their scores, excluding the training facts, and marked all facts that could be logically entailed from the two training facts as correct results and the rest of them as incorrect. We performed 10 runs and in each run we computed the MRR, P@10, MAP for the two models. Finally we averaged these quantities over 10 runs.

Results Table 6 shows that our method was able to rank logically entailed facts with much higher precision and recall than the baseline systems. This validates our intuition that logical rules can be usefully incorporated into the parameter learning mechanism

of a KBC system via simple geometric constraints even for low dimensional embeddings. The reason for the large improvement in performance by the ELKB system in comparison to the baseline is that the ELKB model makes the score of entailed facts higher than the score of non-entailed facts because of the constraints during learning. E.g. the scores of entailed facts such as `(Considered,(Nono,Enemy))`, and `(TransactWith,(Benedict,Nono))` are forced to be high in comparison to non-entailed facts such as `(TradeWith,(Benedict,WMD))`. In comparison the baseline method does not have this systematic advantage and its scores remain unchanged.

4.1 Link Prediction on WordNet

In the link prediction task, the KBC system is given incomplete facts, with either a missing head entity or tail entity, i.e. given either $(r,(\_,e'))$, or $(r,(e,\_))$ the system has to predict e or e' respectively. We evaluated the utility of proposed constraints by comparing the performance of model A and model B trained with and without the constraints. We now present the results of our experiments on the WN18 knowledge graph,[7] derived from WordNet, and released by Bordes et al. 2013, which is a popular testbed for KBC algorithms Wang et al. (2014); Lin et al. (2015); Toutanova et al. (2015); Yang et al. (2015).

Data The WN18 dataset comes with standard train, development and test splits. These three splits of the data contain 141442, 5000, and 5000 facts respectively. The total number of relations in the WN18 dataset is 18 and the number of entities is 40,943. Recently Guo et al. 2016 publicly released a list of logical rules[8] which we directly incorporated into our framework. All of their rules were REVIMP rules.

For all the models we fixed batch size$=10$, $\alpha=0.001$,$\eta=0.125$,$S=200$,$\tilde{d}=100$, for model T, $d=100$ and otherwise $d=200$. Following existing work we measured the MRR, HITS@3 and Hits@10 metrics and report their average over the two tasks of head entity prediction and tail entity prediction. Instead of training in a single pass we trained our models for 50 epochs on the WN18 dataset and chose the best parameters using early stopping on the validation set. In other words, we used the parameters from that epoch which performed the best on the validation set in terms of the HITS@10 metric. Finally, we combined the predictions of the best performing T model and the best performing system based on model B. In order to combine the two ranking systems we trained a logistic regression classifier using the default settings in vowpal webbit[9] to first predict whether model T or model B will produce a better ranking and then output that system's ranking over entities for evaluation. Our logistic regression classifier had 73% accuracy on the training data and 70% accuracy over the test data. By using this third system we were able to create a single ranking system that performed better than model T which is very similar to the TransE model.[10]

Results Table 7 shows that both the constrained and unconstrained versions of model A perform quite poorly. This is to be expected since model A scores a triplet $(r,(e,e'))$ as $\langle\mathbf{r},\mathbf{e}\rangle+\langle\mathbf{r},\mathbf{e}'\rangle$. Regardless of e', the ranking produced by the model will remain the same. Therefore model A is clearly unsuitable for this task, for similar reason model C is also an unsuitable model. However the drastic improvement in the performance of model B when it is trained according to the constraints corresponding to the REVIMP rules demonstrates the utility of our proposed constraints. After adding the constraints, the MRR increased almost 3 times and the value of HITS@10 by 4 times from 0.137. Recall that at test / inference time the constraints do not play any role so the only role of the constraints is as a form of regularization on the parameters of the model.

Model	Project	MRR	HITS@3	HITS@10
A	No	0.0152	0.016	0.0330
A	Yes	0.0238	0.03	0.0514
B	No	0.0677	0.072	0.137
B	Yes	0.241	0.283	0.50
T	No	0.311	0.412	0.66
$B^{project}$+T	–	**0.367**	**0.475**	**0.708**

Table 7: MRR, HITS@3 and HITS@10 of the constrained and unconstrained versions of models A, B and unconstrained T. B+T reports the results of combining models B and model T.

5 Conclusion

We have presented a novel method for incorporating logical constraints into an embedding based knowledge base by constraining the parameters of a KB. Our experiments on a small logical deduction problem, and on WordNet, indicate that our ideas of imposing geometric constraints on embeddings for enforcing logical rules are sound, and that they can improve the generalization of models that are hard to train otherwise. Although the KBC models A, B, C and D do not perform as well as existing models trained without constraints such as TransE, we show that they can be used as part of a combination of systems to improve upon existing methods.

[7] We found that the performance of models C,D and R was too low therefore we do not report their results.

[8] `aclweb.org/anthology/attachments/D/D16/D16-1019.Attachment.zip`

[9] `https://github.com/JohnLangford/vowpal_wabbit`

[10] The main differences between model T and TransE are that TransE used hinge loss versus the BPR objective. TransE does not regularize the relation embeddings and forces the entity embeddings to lie on the unit sphere, instead in model T we add a quadratic regularization term to regularize the embeddings.

References

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795.

Bowman, S. R., Potts, C., and Manning, C. D. (2015a). Learning distributed word representations for natural logic reasoning. In *Proceedings of the AAAI Spring Symposium on Knowledge Representation and Reasoning*.

Bowman, S. R., Potts, C., and Manning, C. D. (2015b). Recursive neural networks can learn logical semantics. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 12–21, Beijing, China. ACL.

Demeester, T., Rocktäschel, T., and Riedel, S. (2016). Lifted rule injection for relation embeddings. In *Proceedings of the EMNLP*, pages 1389–1399, Austin, Texas. ACL.

Garcez, A. S. d., Gabbay, D. M., and Broda, K. B. (2002). *Neural-Symbolic Learning System: Foundations and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Grefenstette, E. (2013). Towards a formal distributional semantics: Simulating logical calculi with tensors. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1*, pages 1–10, Atlanta, Georgia, USA. ACL.

Gruber, P. M. (2007). *Convex and Discrete Geometry*. Springer.

Guo, S., Wang, Q., Wang, B., Wang, L., and Guo, L. (2015). Semantically smooth knowledge graph embedding. In *Proceedings of the ACL*, pages 84–94, Beijing, China. ACL.

Guo, S., Wang, Q., Wang, L., Wang, B., and Guo, L. (2016). Jointly embedding knowledge graphs and logical rules. In *Proceedings of the EMNLP*, pages 192–202, Austin, Texas. ACL.

Hammer, B. and Hitzler, P. (2007). *Perspectives of neural-symbolic integration*, volume 77. Springer.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *The 29th AAAI Conference on Artificial Intelligence*, pages 2181–2187. AAAI.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the UAI*, pages 452–461. AUAI Press.

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). Relation extraction with matrix factorization and universal schemas. In *Proceedings of the ACL*, pages 74–84, Atlanta, Georgia. ACL.

Rocktäschel, T., Bošnjak, M., Singh, S., and Riedel, S. (2014). Low-dimensional embeddings of logic. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 45–49, Baltimore, MD. ACL.

Rocktäschel, T., Singh, S., and Riedel, S. (2015). Injecting logical background knowledge into embeddings for relation extraction. In *NAACL*.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Russell, S., Norvig, P., and Intelligence, A. (1995). A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27.

Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. (2015). Representing text for joint embedding of text and knowledge bases. In *Proceedings of the EMNLP*, pages 1499–1509, Lisbon, Portugal. ACL.

Vendrov, I., Kiros, J. R., Fidler, S., and Urtasun, R. (2016). Order-embeddings of images and language. In *Proceedings of the International Conference on Learning Representations*.

Wang, W. Y. and Cohen, W. W. (2016). Learning first-order logic embeddings via matrix factorization. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, New York, NY. AAAI.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119. AAAI Press.

Yang, B., Yih, W.-t., 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*.