Journal of Intelligent Information Systems

Efficient Energy-Based Embedding Models for Link Prediction in Knowledge Graphs

--Manuscript Draft--

Manuscript Number:	
Full Title:	Efficient Energy-Based Embedding Models for Link Prediction in Knowledge Graphs
Article Type:	Recent Advances in Mining Patterns from Complex Data
Keywords:	Energy-Based Embedding Models; Link Predictions; RDF Knowledge Graphs
Corresponding Author:	Claudia d'Amato University of Bari Bari, ITALY
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	University of Bari
Corresponding Author's Secondary Institution:	
First Author:	Pasquale Minervini
First Author Secondary Information:	
Order of Authors:	Pasquale Minervini
	Claudia d'Amato
	Nicola Fanizzi
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	We focus on the problem of link prediction in Knowledge Graphs with the goal of discovering new facts. To this purpose, Energy-Based Models for Knowledge Graphs that embed entities and relations in continuous vector spaces have largely been used. The main limitation on their applicability lies in the parameter learning phase that may require a large amount of time for converging to optimal solutions. For this reason we propose a unified view of the Energy-Based Embedding Models that is grounded on an adaptive learning rate, showing that this kind of selection can improve the efficiency of the parameter learning process by an order of magnitude with respect to existing methods, leading to more accurate link prediction models in a significantly lower number of iterations. Finally, we also employ the proposed learning procedure for evaluating a variety of new models. Our results show a significant improvement over state-of-the-art link prediction methods on two large knowledge graphs: WordNet and Freebase.

Noname manuscript No.
(will be inserted by the editor)

Efficient Energy-Based Embedding Models for Link Prediction in Knowledge Graphs

Pasquale Minervini · Claudia d'Amato · Nicola Fanizzi

the date of receipt and acceptance should be inserted later

Abstract We focus on the problem of link prediction in Knowledge Graphs with the goal of discovering new facts. To this purpose, Energy-Based Models for Knowledge Graphs that embed entities and relations in continuous vector spaces have largely been used. The main limitation on their applicability lies in the parameter learning phase that may require a large amount of time for converging to optimal solutions. For this reason we propose a unified view of the Energy-Based Embedding Models that is grounded on an adaptive learning rate, showing that this kind of selection can improve the efficiency of the parameter learning process by an order of magnitude with respect to existing methods, leading to more accurate link prediction models in a significantly lower number of iterations. Finally, we also employ the proposed learning procedure for evaluating a variety of new models. Our results show a significant improvement over state-of-the-art link prediction methods on two large knowledge graphs: WordNet and Freebase.

1 Introduction

Knowledge Graphs (KGs) are graph-structured knowledge bases, where factual knowledge is represented in the form of relationships between entities. We focus on KGs that adopt *Resource Description Framework* (RDF)[1] as their representation, since they constitute a powerful instrument for search, analytics, recommendations, and data integration. Indeed, RDF is the Web standard for expressing information about resources.

A resource (hereafter also called *entity*) can be anything, including documents, people, physical objects, and abstract concepts. An RDF knowledge

Department of Computer Science - University of Bari, Italy
E-mail: {firstname.lastname}@uniba.it

[1] http://www.w3.org/TR/rdf11-concepts/

base (also called *RDF graph* as a KG) is a set of *RDF triples* in the form $\langle s, p, o \rangle$, where s, p and o denote the *subject*, the *predicate* and the *object* of the triple, respectively. The s and o position are generally intended as *resources* while p denotes a *predicate* (i.e. a *relation type*). Each triple $\langle s, p, o \rangle$ describes a statement, which can be interpreted as: *A relationship of type p holds between entities s and o.* The following example shows a set of RDF triples[2] describing the writer *William Shakespeare*[3]:

Example 1 (RDF Fragment)

⟨W. Shakespeare,	influencedBy,	G. Chaucer⟩
⟨W. Shakespeare,	religion,	Church of England⟩
⟨W. Shakespeare,	author,	Hamlet⟩
⟨Hamlet,	genre,	Tragedy⟩
⟨Hamlet,	character,	Ophelia⟩

Several RDF KGs are publicly available through the *Linked Open Data* (LOD) cloud, a collection of interlinked KGs such as Freebase [4], DBpedia [3] and YAGO [18]. As of April 2014, the LOD cloud is composed by 1,091 interlinked KGs, globally describing more than 8×10^6 entities, and 188×10^6 relationships holding between them[4]. However, KGs are often largely incomplete. For instance, 71% of the persons described in Freebase[5] have no known place of birth and 75% of them have no known nationality [11].

For this reason, in this work, we focus on the problem of *predicting missing links* in large KGs, so to discover new facts about a domain of interest. In the literature, this problem is referred to as *link prediction*, or *knowledge base completion*. The aim of this work is to provide an efficient and accurate model for predicting missing RDF triples in large RDF KGs (in a *link prediction* setting), without requiring extra background knowledge.

Specifically, we focus on a class of Energy-Based Models for KGs, where entities and relations are embedded in continuous vector spaces, referred to as *embedding spaces*. In such models, the probability of an RDF triple is expressed in terms of *energy* of the triple, i.e. an unnormalized measure that is inversely proportional to the triple probability and is computed as a function of the embedding vectors of the subject, the predicate and the object of the triple. In the following, we refer to models in this class as *Energy-Based Embedding Models* (EBEMs). These models, such as *Translating Embedding* (TransE) [6] and other related ones [5,7,25], achieve state-of-the-art results on link prediction tasks, while being able to scale to large KGs, such as WordNet and Freebase. However, a major limiting factor lies in the parameter

[2] This description is taken from the Freebase KG [4]

[3] For readability reasons, we describe entities and relations using abbreviated forms rather than the pure RDF syntax.

[4] State of the LOD Cloud 2014: `http://lod-cloud.net/`

[5] Available at `https://developers.google.com/freebase/data`

learning algorithm, which may require a long time (even days) to converge on large KGs [9].

In order to overcome such a limitation, we propose a method for reducing the learning time in EBEMs by an order of magnitude, while leading to more accurate link prediction models. Furthermore, we employ the proposed learning method for evaluating a family of novel EBEMs with useful properties. We experimentally tested our methods on two large and commonly used KGs: namely WordNet and Freebase by showing a significant improvement over state-of-the-art link prediction methods. on two large knowledge graphs:, namely WordNet and Freebase.

The rest of the paper is organized as follows. In Sect. 2 we introduce basics on Energy-Based Models and their application to RDF KGs. In Sect. 3 we propose a framework for characterizing state-of-the-art EBEMs, together with a family of novel energy functions with useful properties, and a method for improving the efficiency of the learning process in such models. In Sect. 4 we survey related works. In Sect. 5 we empirically evaluate the proposed learning methods and energy functions. In Sect. 6 we summarize this work, and outline future research directions.

2 Basics

In this section we summarize the basics of Energy-Based Models, after that we focus on the formalization of Energy-Based Models for RDF KGs.

2.1 Energy-Based Models

Energy-Based Models (EBMs) [17] are a versatile and flexible framework for modeling dependencies between variables. The key component in EBMs is a scalar-valued *energy function* $E(\cdot)$, which associates a scalar *energy* to a configuration of variables. The energy of a configuration of variables is inversely proportional to the probability of the configuration of variables: more likely configurations correspond to lower energy values, while less likely configurations correspond to higher energy values. Two main steps can be recognized in EBMs: the *inference* step and the *learning* step.

In EBMs, *inference* consists in finding the most likely configuration of the variables of interest, that is the one that minimizes the energy function $E(\cdot)$. Let X and Y be random variables, with values in $\mathcal{X}$ and $\mathcal{Y}$. An example of the exploitation of the inference in EBMs is given in the following.

Example 2 (Energy-Based Inference) Assume that X describes the pixels of an image, while Y describes a discrete label associated to the image (such as “car” or “tree”). Let $E : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ be an energy function defined on the configurations of X and Y. The most likely label $y^* \in \mathcal{Y}$ for an image $x \in \mathcal{X}$

can be inferred by finding the label in $\mathcal{Y}$ that, given x, minimizes the energy function $E(\cdot)$:

$$y^* = \arg\min_{y \in \mathcal{Y}} E(x, y).$$

Learning in EBMs consists in finding the most appropriate energy function within a family $\mathcal{F} = \{E_\theta \mid \theta \in \Theta\}$, indexed by parameters θ. That is in finding the function that is actually able to associates *lower energy states* to likely configurations of the variables of interests, and *higher energy states* to unlikely configurations of such variables. In practice, this corresponds to finding the energy function $E^*_\theta \in \mathcal{F}$ that minimizes a given *loss functional* $\mathcal{L}$, which measures the *quality* of the energy function on the data $\mathcal{D}$:

$$E^*_\theta = \arg\min_{E_\theta \in \mathcal{F}} \mathcal{L}(E_\theta, \mathcal{D}).$$

A normalized probability distribution can be derived from an EBM. Specifically, given an energy function $E : \mathcal{X} \mapsto \mathbb{R}$ defined on the possible configurations of a random variable X, it is possible to derive a corresponding probability distribution through the *Gibbs distribution*:

$$P(X = x) = \frac{1}{Z(\beta)} e^{-\beta E(x)},$$

where β is an arbitrary positive constant, and $Z(\beta) = \sum_{\tilde{x} \in \mathcal{X}} e^{-\beta E(\tilde{x})}$ is a normalizing factor [6] referred to as the *partition function.*

2.2 Energy-Based Models for RDF KGs

EBMs can be used for modeling the uncertainty in RDF KGs, in both statistical inference and learning tasks.

An RDF graph G can be viewed as a labeled directed multigraph, where entities are vertices, and each RDF triple is represented by a directed edge whose label is a predicate, and emanating from its source vertex to its object vertex. We denote as $\mathcal{E}_G$ the set of all entities occurring as subjects or objects in G, that is $\mathcal{E}_G = \{s \mid \exists \langle s, p, o \rangle \in G\} \cup \{o \mid \exists \langle s, p, o \rangle \in G\}$, and as $\mathcal{R}_G$ the set of all relations appearing as predicates in G, that is $\mathcal{R}_G = \{p \mid \exists \langle s, p, o \rangle \in G\}$. Let $\mathcal{S}_G = \mathcal{E}_G \times \mathcal{R}_G \times \mathcal{E}_G$ be the space of *possible triples* of G, with $G \subseteq \mathcal{S}_G$, and let $E : \mathcal{S}_G \mapsto \mathbb{R}$ be an energy function that defines an energy distribution over the set of possible triples $\mathcal{S}_G$. The most likely object $o^* \in \mathcal{E}_G$ to appear in a RDF triple with subject $s \in \mathcal{E}_G$ and predicate $p \in \mathcal{R}_G$, can be determined according to $E(\cdot)$. Specifically, the most likely object o^* can be inferred by finding the object $o \in \mathcal{E}_G$ that minimizes $E(\cdot)$:

$$o^* = \arg\min_{o \in \mathcal{E}_G} E(\langle s, p, o \rangle).$$

[6] If X is a continuous random variable, then $Z(\beta) = \int_{\tilde{x} \in \mathcal{X}} e^{-\beta E(\tilde{x})}$.

As mentioned in Sect. 1, in this work we will focus on *Energy-Based Embedding Models* (EBEMs) (presented in Sect. 3) that are a specific class of EBMs for RDF KGs where each entity $x \in \mathcal{E}_G$ is mapped to a unique low-dimensional continuous vector $\mathbf{e}_x \in \mathbb{R}^k$ that is referred to as the *embedding vector* of x. The reason for such a choice is that EBEMs, such as *Translating Embedding* (TransE) [6] and related models [5, 7, 25], achieve performances that are comparable with state-of-the-art link prediction methods, while scaling to large RDF KGs such as WordNet and Freebase.

3 A Framework for Energy-Based Embedding Models

In this section, we present a general framework for formalizing EBEMs for KGs in a unified view.

Given an RDF graph G with entities $\mathcal{E}_G$, relations $\mathcal{R}_G$, and $\mathcal{S}_G = \mathcal{E}_G \times \mathcal{R}_G \times \mathcal{E}_G$ the space of possible triples of G, similarly to EBMs, an EBEM associates an *energy* value to each triple in $\mathcal{S}_G$, by means of an energy function $E_\theta : \mathcal{S}_G \rightarrow \mathbb{R}$, with parameters θ. As for EBMs, *Learning* in EBEMs consists in finding an energy function $E_\theta^* \in \mathcal{F}$, within a parametric family of energy functions $\mathcal{F} = \{E_\theta \mid \theta \in \Theta\}$ indexed by parameters θ, that minimizes a given loss functional $\mathcal{L}$ defined on the RDF graph G:

$$E_\theta^* = \arg \min_{E_\theta^* \in \mathcal{F}} \mathcal{L}(E_\theta, G).$$

Since the *energy* value for a triple expresses a quantity that is inversely proportional to the probability of the triple itself (see Sect. 2), in a *link prediction* setting, the energy function $E_\theta^*(\cdot)$ can be exploited for assessing a ranking of the so called *unobserved* triples, that are the triples in $\mathcal{S}_G \setminus G$. As such, triples associated to lower energy values (higher probabilities) will be more likely to be considered for a completion of the graph G. Indeed, in RDF, the *Open World Assumption* holds, which means that a missing triple in G does not mean that the corresponding statement is false (like for the case of the *Closed World Assumption* typically made in the database setting), but rather that its truth value is *missing/unknown* since it cannot be observed in the KG. We refer to all triples in G as *visible triples*, and to all triples in $\mathcal{S}_G \setminus G$ as *unobserved triples*, which might encode true statements.

In the following sections, we show that EBEMs proposed in literature so far can be characterized with respect to their energy function and we also propose novel formulations of the energy functions with useful properties (see Sect. 3.1). Hence we focus on the *learning process*, by specifically proposing a method for improving the efficiency of the parameters learning step (see Sect. 3.2).

Table 1: Energy-Based Embedding Models for knowledge graphs proposed in literature, with their energy functions, shared and embedding parameters.

Model	Energy function $E(\langle s,p,o\rangle)$	Shared	Embedding
Unstructured [5]	$\lVert \mathbf{e}_s - \mathbf{e}_o \rVert_1$		$\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$
TransE [6]	$\lVert (\mathbf{e}_s + \mathbf{e}_p) - \mathbf{e}_o \rVert_{1/2}$		$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{R}^k$
SE [7]	$\lVert \mathbf{R}_{p,1}\mathbf{e}_s - \mathbf{R}_{p,2}\mathbf{e}_o \rVert_1$		$\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k, \mathbf{R}_{p,\cdot} \in \mathbb{R}^{n \times k}$
RESCAL [21]	$\mathbf{e}_s^T \mathbf{R}_p \mathbf{e}_o$		$\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k, \mathbf{R} \in \mathbb{R}^{k \times k}$
SME lin. [5]	$(\mathbf{R}_1\mathbf{e}_s + \mathbf{R}_2\mathbf{e}_p)^T(\mathbf{R}_3\mathbf{e}_o + \mathbf{R}_4\mathbf{e}_p)$	$\mathbf{R}_\cdot \in \mathbb{R}^{n \times k}$	$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{R}^k$
SME bil. [5]	$[(\mathbf{R}_1\mathbf{e}_s) \times_3 (\mathbf{R}_2\mathbf{e}_p)]^T\,[(\mathbf{R}_3\mathbf{e}_o) \times_3 (\mathbf{R}_4\mathbf{e}_p)]$	$\mathbf{R}_\cdot \in \mathbb{R}^{n \times k}$	$\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{R}^k$
NTN [25]	$\mathbf{u}_p^T \tanh\left(\mathbf{e}_s^T \mathbf{T}_p \mathbf{e}_o + \mathbf{R}_{p,1}\mathbf{e}_s + \mathbf{R}_{p,2}\mathbf{e}_o\right)$		$\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k, \mathbf{u}_p \in \mathbb{R}^n,$ $\mathbf{T}_p \in \mathbb{R}^{k \times k \times n}, \mathbf{R}_{p,\cdot} \in \mathbb{R}^{n \times k}$

3.1 Energy Function Characterization and New energy Functions

The energy function $E_\theta : \mathcal{S}_G \rightarrow \mathbb{R}$ of state-of-the art EBEMs for KG has two types of parameters:

- **Shared Parameters:** used for computing the energy of all triples in the space of possible triples $\mathcal{S}_G$ of G.
- **Embedding Parameters:** used for computing the energy of triples containing a specific entity or relation $x \in \mathcal{E}_G \cup \mathcal{R}_G$. We denote such parameters by adding a suffix with the name of the entity or relation they are associated to (e.g. $\mathbf{e}_s$ denotes the embedding vector of s).

EBEMs for KGs associate each entity $x \in \mathcal{E}_G$ to a k-dimensional embedding vector $\mathbf{e}_x \in \mathbb{R}^k$, and each relation $p \in \mathcal{R}_G$ to a (possibly empty) set of embedding parameters $\mathbf{S}_p$. Tab. 1 summarizes the energy functions of state-of-the-art EBEMs for KGs, by highlighting the distinction between the two different kinds of parameters reported above.

The energy functions can be seen as sharing a common structure: given a RDF triple $\langle s,p,o\rangle$, its energy $E(\langle s,p,o\rangle)$ is computed by the following two steps process:

1. The embedding vectors $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$, of the subject s and the object o of the triple, and the embedding parameters $\mathbf{S}_p$ associated to the predicate p of the triple are used to obtain two new vectors $\mathbf{e}'_s, \mathbf{e}'_o \in \mathbb{R}^{k'}$ by means of two functions $f_s(\cdot)$ and $f_o(\cdot)$ (see also Fig. 1):
$$\mathbf{e}'_s = f_s(\mathbf{e}_s, \mathbf{S}_p), \qquad \mathbf{e}'_o = f_o(\mathbf{e}_o, \mathbf{S}_p).$$
2. The energy of $\langle s,p,o\rangle$ is computed by a function $g : \mathbb{R}^{k'} \times \mathbb{R}^{k'} \mapsto \mathbb{R}$, applied to the vectors $\mathbf{e}'_s, \mathbf{e}'_o \in \mathbb{R}^{k'}$ resulting from the previous step:
$$E(\langle s,p,o\rangle) = g(\mathbf{e}'_s, \mathbf{e}'_o) = g(f_s(\mathbf{e}_s, \mathbf{S}_p), f_o(\mathbf{e}_o, \mathbf{S}_p)). \tag{1}$$

The two steps process for computing the energy function for an EBEM is clearly depicted in Fig. 1. As an example, in the following we show how the energy function adopted by the *Translating Embeddings* model (TransE) [6], which stands for the main state of the art EBEM for performing link prediction in KG, can be expressed by the use of the formalization presented just above. Specifically, TransE is particularly interesting since while its number of parameters grows *linearly* with the number of entities and relations in the knowledge graph, it yields state-of-the-art link prediction results on the WordNet and Freebase knowledge graphs (see the empirical comparison with other link prediction methods in Sect. 5.1).

$$\begin{tikzcd}
\mathbf{e}_s \arrow[d] & \mathbf{S}_p \arrow[dl] \arrow[dr] & \mathbf{e}_o \arrow[d] \\
\mathbf{e}'_s = f_s(\mathbf{e}_s, \mathbf{S}_p) \arrow[dr] & & \mathbf{e}'_o = f_o(\mathbf{e}_o, \mathbf{S}_p) \arrow[dl] \\
& E(\langle s, p, o \rangle) = g(\mathbf{e}'_s, \mathbf{e}'_o) &
\end{tikzcd}$$

Fig. 1: Structure of the energy function in Energy-Based Embedding Models for KGs: $\mathbf{e}_s$, $\mathbf{S}_p$ and $\mathbf{e}_o$ are the embedding parameters of s, p and o.

Example 3 (Energy Function in TransE) In the formulation for the energy function of the *Translating Embeddings* model (TransE) [6] (see Tab.1), each entity and relation $x \in \mathcal{E}_G \cup \mathcal{R}_G$ in an RDF graph G is associated (correspond) to an embedding vector $\mathbf{e}_x \in \mathbb{R}^k$ in the embedding space, while each relation corresponds to a *translation operation* in such an embedding space. As from Tab. 1, the energy function can be formulated by using the L_1 or the L_2 norm. In the case of L_1 formulation, the *energy* of an RDF triple $\langle s, p, o \rangle$ is given by the L_1 distance of $(\mathbf{e}_s + \mathbf{e}_p)$ and $\mathbf{e}_o$:

$$E(\langle s, p, o \rangle) = \| (\mathbf{e}_s + \mathbf{e}_p) - \mathbf{e}_o \|_1.$$

This corresponds to the following choice of the functions $f_s(\cdot)$, $f_o(\cdot)$ and $g(\cdot)$:

$$f_s(\mathbf{e}_s, \{\mathbf{e}_p\}) = \mathbf{e}_s + \mathbf{e}_p, \qquad f_o(\mathbf{e}_o, \{\mathbf{e}_p\}) = \mathbf{e}_o, \qquad g(\mathbf{e}'_s, \mathbf{e}'_o) = \|\mathbf{e}'_s - \mathbf{e}'_o\|_1.$$

□

Besides of proposing a general framework for expressing an energy function to be used by EBEMs, in this work, we also investigate whether the choice of other *affine transformations* for the functions $f_s(\cdot)$ and $f_o(\cdot)$, such as *scaling*, or *composition of translation and scaling*, leads to more accurate models than those generated by TransE (using the energy function reported in Tab. 1 and reformulated as shown just above), while still having a number of parameters that scales linearly in the number of entities and relations. Specifically, we investigate the following choices for the functions $f_s(\cdot)$ and $f_o(\cdot)$:

Translation:	$f(\mathbf{e}_x, \{\mathbf{e}_p\}) = \mathbf{e}_x + \mathbf{e}_p,$
Scaling:	$f(\mathbf{e}_x, \{\mathbf{e}_p\}) = \mathbf{e}_x \odot \mathbf{e}_p,$
Scaling ∘ Translation:	$f(\mathbf{e}_x, \{\mathbf{e}_{p,1}, \mathbf{e}_{p,2}\}) = (\mathbf{e}_x \odot \mathbf{e}_{p,1}) + \mathbf{e}_{p,2},$

where $\circ$ denotes the composition operation between functions, and $\odot$ denotes the Hadamard product. The results of such a study are reported and discussed in Sect. 5.1. Please note that, similarly to models in [5–7], we enforce the embedding vector of all entities to lie on the Euclidean unit $(k-1)$-sphere, that is $\mathbb{S}^{k-1} = \{\mathbf{x} \in \mathbb{R}^k \mid \|\mathbf{x}\|_2 = 1\}$ (see Alg. 1, line 3). For such a reason, we also propose normalizing the results of functions $f_s(\cdot)$ and $f_o(\cdot)$, so the resulting projections also lie on the Euclidean unit sphere.

In the next section we focus on the *learning* step of EBEMs, consisting (as illustrated in Sect. 2) in finding the most appropriate energy function to be used for the successive *inference* step.

3.2 Learning the Parameters of the Energy Function

As illustrated in Sect. 2, *learning* in EBEMs for KGs corresponds to finding an energy function E^*_θ, within a family of functions $\mathcal{F} = \{E_\theta \mid \theta \in \Theta\}$ indexed by parameters θ, that minimizes a given *loss functional* $\mathcal{L}$ measuring the *quality* of an energy function with respect to the RDF graph G:

$$E^*_\theta = \arg\min_{E_\theta \in \mathcal{F}} \mathcal{L}(E_\theta, G).$$

In the following, the definition for the *loss functional* $\mathcal{L}$ is given. In agreement with the formalization given in Sect. 3.1, a key point for learning the (best) energy function in EBEMs consists in learning the shared and embedding parameters to be used for computing the energy function. As in [5–7], shared and embedding parameters are learned by using a *corruption process* $\mathcal{Q}(\tilde{x} \mid x)$ that, given a RDF triple $x \in G$, produces a *corrupted* RDF triple $\tilde{x}$, uniformly sampled from the set of corrupted triples $\mathcal{C}_x$. Formally, given an RDF triple $\langle s, p, o \rangle$ from G, the set of corrupted triples for it is given by

$$\mathcal{C}_{\langle s,p,o \rangle} = \{\langle \tilde{s}, p, o \rangle \mid \tilde{s} \in \mathcal{E}_G\} \cup \{\langle s, p, \tilde{o} \rangle \mid \tilde{o} \in \mathcal{E}_G\}$$

that is the set obtained by replacing either the subject or the object of the triple with another entity from the set of entities $\mathcal{E}_G$. The corruption process $\mathcal{Q}(\tilde{x} \mid x)$ is used for defining the following margin-based stochastic ranking criterion over triples in G:

$$\mathcal{L}(E_\theta, G) = \sum_{x \in G} \sum_{\tilde{x} \sim \mathcal{Q}(\tilde{x} \mid x)} \left[\gamma + E_\theta(x) - E_\theta(\tilde{x})\right]_+ , \qquad (2)$$

where $[x]_+ = \max\{0, x\}$, and $\gamma > 0$ is a hyperparameter referred to as *margin.*

As proposed [5–7], the minimization of the loss functional in Eq. 2 can be carried out by projected Stochastic Gradient Descent (SGD) in mini-batch mode, as summarized in Alg. 1. Given an RDF graph G, at each iteration, the algorithm samples a batch of triples from G. Similarly to [6], each batch is obtained by first randomly permuting all triples in G, then partitioning them into n_b batches of similar size, and iterating over them. A single pass over all triples in G is called an *epoch.* For each triple in the batch, the algorithm generates a *corrupted* triple by means of the corruption process $\mathcal{Q}(\tilde{x} \mid x)$:

Algorithm 1 Learning in EBEMs via *Stochastic Gradient Descent* [6]

```
Input: Learning rate $\eta$, batch size $n$
Output: Optimal model parameters $\theta^*$
1: Initialize model parameters $\theta_0$
2: for $t \in \langle 1, \ldots, \tau \rangle$ do
3:     $\mathbf{e}_x \leftarrow \mathbf{e}_x / \|\mathbf{e}_x\|, \forall x \in \mathcal{E}_G$          {Normalize all entity embeddings}
4:     $T \leftarrow \textsc{SampleBatch}(G, n)$          {Sample observed and corrupted triples}
5:     $g_t \leftarrow \nabla \sum_{(x,\tilde{x}) \in T} [\gamma + E_\theta(x) - E_\theta(\tilde{x})]_+$          {Evaluate the gradient of $\mathcal{L}$ w.r.t. $\theta$}
6:     $\Delta_t \leftarrow -\eta g_t$          {Calculate the update to model parameters $\theta$}
7:     $\theta_t \leftarrow \theta_{t-1} + \Delta_t$          {Update the model parameters $\theta$}
8: end for
9: return $\theta_\tau$
```

this leads to a set of observed and corrupted pairs of triples T. Then, the observed/corrupted triple pairs in T are used to evaluate the gradient of the loss functional $\mathcal{L}$ in Eq. 2 with respect to the current model parameters θ. Finally, θ is updated in the steepest descent direction of the loss functional $\mathcal{L}$ by a fixed learning rate η. This procedure is repeated until convergence (in [6] the learning procedure was limited to 1000 epochs).

The main drawback of SGD is that it requires an initial, careful tuning of the learning rate η that is also used across all parameters, without adapting to the characteristics of each parameter. However, if some entities and relations are infrequent, the corresponding embedding parameters will tend to be updated less frequently during the learning process. For such a reason, the task of learning the model parameters in EBEMs by using SGD may require days to terminate [9].

In order to reduce the learning time in EBEMs, we propose the adoption of *adaptive per-parameter learning rates*. Specifically, while the SGD algorithm in Alg. 1 uses a global, fixed learning rate η, we propose relying on methods that estimate the optimal learning rate for each parameter, while still being tractable for learning large models. In particular, we consider the following criteria for selecting the optimal learning rates: the Momentum method [23], AdaGrad [12] and AdaDelta [28]. Each of these methods can be implemented in Alg. 1, by replacing the update to model parameters on line 6 as specified in the following.

Momentum Method The basic idea of this method is accelerating the progress along dimensions where the sign of the gradient does not change, while slowing the progress along dimensions where the sign of the gradient continues to change. This is done by keeping track of previous parameter updates with an exponential decay. The update step on line 6 of Alg. 1, in the Momentum method is given by:

$$\Delta_t \leftarrow \rho \Delta_{t-1} - \eta g_t,$$

where ρ is a hyperparameter controlling the decay of previous parameter updates.

AdaGrad The underlying idea in this method is that per parameter learning rates should grow with the inverse of gradient magnitudes: large gradients should have smaller learning rates, while small gradients should have larger learning rates, so that the progress along each dimension evens out over time. The update step on line 6 of Alg. 1, in AdaGrad is given by:

$$\Delta_t \leftarrow -\frac{\eta}{\sqrt{\sum_{j=1}^{t} g_j^2}} g_t,$$

where η is a global scaling hyperparameter. AdaGrad has been used on large scale learning tasks in a distributed environment [10].

AdaDelta This method uses an exponentially decaying average of squared gradients $E[g^2]$ and squared updates $E[\Delta^2]$, controlled by a decay term ρ, to give more importance to more recent gradients and updates. The update step on line 6 of Alg. 1, in AdaDelta is given by:

$$\Delta_t \leftarrow -\frac{\mathrm{RMS}[\Delta]_{t-1}}{\mathrm{RMS}[g]_t} g_t,$$

where $\mathrm{RMS}[x]_t = \sqrt{E[x^2]_t + \epsilon}$, and ϵ is an offset hyperparameter.

4 Related work

Statistical Relational Learning (SRL) [13] aims at modeling data from multi-relational domains, such as social networks, citation networks, protein interaction networks and knowledge graphs. and one of the main goals is link prediction in such relational domains. Two main categories of models can be ascribed to SRL: *Probabilistic latent variable models* and *Embedding Models* (see Fig. 2 for their main characteristics). The main related works falling in these categories are analyzed in the following.

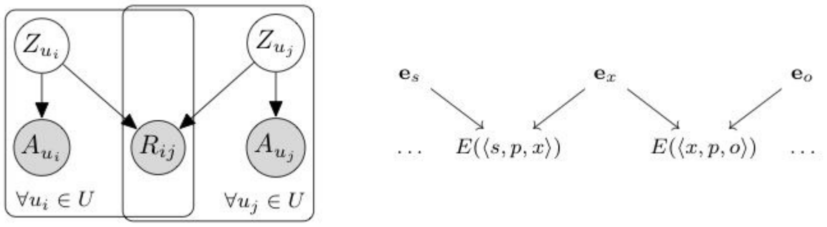

Fig. 2: Left – A simple SB for a social network: each user $u \in U$ is associated to a latent class variable Z_u which conditions both its attributes A_u, and its relations with other users. Right – An example of EBEM: the embedding vector $\mathbf{e}_x$ of an entity x defines the *energy* of all RDF triples in which x appears in.

Probabilistic Latent Variable Models Models in this class explain relations between entities by associating each entity to a set of intrinsic *latent attributes*. The term *latent* refers to the fact that the attributes are not directly observable in the data. Specifically, this class of models condition the probability distribution of the relations between two entities on the *latent attributes* of such entities. Similarly to Hidden Markov Models [16], this allows the information to *propagate* through the network of interconnected latent variables. An early model in this family is the *Stochastic Block Model* (SB) [26], which associates a *latent class* variable to each entity. The *Infinite (Hidden) Relational Model* [15, 27] extends the SB by using Bayesian nonparametrics, so to automatically infer the optimal number of latent classes. The *Infinite Hidden Semantic Model* [22] further extends such model, so to make use of constraints expressed in First Order Logic during the learning process, while the Mixed Membership Stochastic Block Model [1] extends the SB so to allow entities to have mixed cluster-memberships. More recent works associate a set of *latent features* to each entity, instead of a single latent class. The *Nonparametric Latent Feature Relational Model* [20] is a latent feature model, which relies on Bayesian nonparametrics to automatically infer the optimal number of latent features during learning.

The main limitation of probabilistic latent variable models lies in the complexity of probabilistic inference and learning, which is intractable in general [16]. As a consequence, these models may not be feasible for modeling large knowledge graphs.

Embedding Models Similarly to probabilistic latent feature models (see Fig. 2), in *Embedding Models* each entity in the knowledge graph is represented by means of a continuous *embedding vector* $\mathbf{e}_x \in \mathbb{R}^k$, encoding its intrinsic latent features within the KG. Nevertheless, models in this class do not necessarily rely on probabilistic inference for learning the optimal embedding vectors and this allows avoiding the issues related to the proper normalization of probability distributions, that may lead to intractable problems.

In RESCAL [21], the problem of learning the embedding vector representations of all entities and predicates is cast as a *tensor factorization* problem: by relying on a bilinear model, and by using a squared reconstruction loss, its authors propose an efficient learning algorithm based on regularized *Alternating Least Squares*. However, in RESCAL, the number of parameters grows *super-linearly* with the number of predicates in the knowledge graph: for such a reason, RESCAL can hardly scale to highly-relational knowledge graphs [14].

In EBEMs, the *energy* of each RDF triple $\langle s, p, o \rangle$ is defined as a functions of the embedding vectors $\mathbf{e}_s$ and $\mathbf{e}_o$, associated to the subject s and the object o of the triple. The major limitation in EBEMs is the *learning time*, i.e. the time required for learning the parameters of the energy function.

Several options have been proposed for the choice of both the *energy function* and the *loss functional* for learning the embedding vectors representation, e.g. see [5–7, 14, 25]. These methods have been used to achieve state-of-the-art link prediction results while scaling on large KGs. We outperform such methods both in terms of efficiency (for learning the model parameters, reducing the learning time by an order of magnitude) and effectiveness (by obtaining a more accurate model) (see Sect. 5).

5 Empirical Evaluation

In this section, we present the empirical evaluation for our proposed solution. Particularly, we aim at answering the following questions:

Q1: Can adaptive learning rates, as proposed in Sect. 3.2, be used for improving the efficiency of parameters learning with respect to the current state-of-the-art EBEMs?

Q2: Do the energy functions proposed in Sect. 3.1 lead to more accurate link prediction models for knowledge graph completion?

In Sect. 5.1, we answer **Q1** by empirically evaluating the efficiency of the proposed learning procedure, and the accuracy of the learned models. In Sect. 5.2, we answer **Q2** by evaluating the accuracy of models using the proposed energy functions in link prediction tasks.

In the following, we describe the KGs used for the evaluation, jointly with the adopted metrics. Specifically, for comparison purposes, we adopt the same evaluation settings used in [6].

Knowledge Graphs As KGs, WordNet [19] and Freebase (FB15k) [4] have been adopted.

WordNet is a lexical ontology for the English language. It is composed by over 151×10^3 triples, describing 40943 entities and their relations by means of 18 predicate names.

Freebase (FB15k) is a large collaborative knowledge base that is composed by over 592×10^3 triples, describing 14951 entities and their relations by means of 1345 predicate names.

As for the training/validation/test sets, we use the same sets as used in [6]. Specifically, as regards WordNet, given the whole KG, 5000 triples have been removed for validation and 5000 have been used for testing. As regards FB15k, 50000 triples have been removed for validation while 59071 have been used for testing.

Evaluation Metrics In agreement with [6], the following metrics have been used: *averaged rank* (denoted as Mean Rank), and *proportion of ranks not larger than* 10 (denoted as Hits@10). They have been computed as follows.

For each test triple $\langle s, p, o \rangle$, the object o is replaced by each entity $\tilde{o} \in \mathcal{E}_G$ in G thus generating a *corrupted* triple $\langle s, p, \tilde{o} \rangle$. The energy values of corrupted triples are computed by the model, and successively sorted in ascending order. The rank of the correct triple is finally stored. Similarly, this procedure is repeated by corrupting the subject s of each test triple $\langle s, p, o \rangle$. Aggregated over all test triples, this procedure leads to the two metrics: *averaged rank* (denoted as Mean Rank), and *proportion of ranks not larger than* 10 (denoted as Hits@10). This is referred to as the Raw setting.

Please note that, if a generated corrupted triple already exists in the KG, ranking it before the original triple $\langle s, p, o \rangle$ is not wrong. For such a reason, an alternative setting, referred to as the Filtered setting (abbreviated with Filt.) is also considered. In this setting, corrupted triples that exist in either training, validation or test set are removed, before computing the rank of each triple.

In both Raw and Filtered settings, it would be desirable to have low Mean Rank and high Hits@10.

5.1 Evaluation of Adaptive Learning Rates

In order to reply to question **Q1**, that is, for assessing whether Momentum, AdaGrad and AdaDelta are more efficient than SGD in minimizing the loss functional in Eq. 2, we empirically evaluated such methods on the task of learning the parameters in TransE on WordNet and Freebase (FB15k) KGs, using the optimal settings described in [6] that is:

- $k = 20$, $\gamma = 2$, $d = L_1$ for WordNet
- $k = 50$, $\gamma = 1$, $d = L_1$ for FB15k.

Following the empirical comparison of optimization methods in [24], we compared SGD, Momentum, AdaGrad and AdaDelta using an extensive grid of hyperparameters. Specifically, given $\mathcal{G}_\eta = \{10^{-6}, 10^{-5}, \ldots, 10^{1}\}$, $\mathcal{G}_\rho = \{1 - 10^{-4}, 1 - 10^{-3}, \ldots, 1 - 10^{-1}, 0.5\}$ and $\mathcal{G}_\epsilon = \{10^{-6}, 10^{-3}\}$, the grids of hyperparameters for each of the optimization methods were defined as follows:

- **SGD** and **AdaGrad:** rate $\eta \in \mathcal{G}_\eta$.
- **Momentum:** rate $\eta \in \mathcal{G}_\eta$, decay rate $\rho \in \mathcal{G}_\rho$.
- **AdaDelta:** decay rate $\rho \in \mathcal{G}_\rho$, offset $\epsilon \in \mathcal{G}_\epsilon$.

For each possible combination of optimization method and hyperparameter values, we performed an evaluation consisting in 10 learning tasks, each time using a different random seed for initializing the model parameters in TransE. The same 10 random seeds were used for each of the evaluation tasks.

Fig. 3 shows the behavior of the loss function for each of the optimization methods, for the best hyperparameter settings after 100 epochs over the training set. It is immediate to see that, in both WordNet and FB15k knowledge graphs, AdaGrad (with $\eta = 0.1$) and AdaDelta (with $(1 - \rho) = 10^{-3}$ and $\epsilon = 10^{6}$) provide sensibly lower values of the loss functional $\mathcal{L}$ than SGD and Momentum, even after a low number of iterations ($<$ 10 epochs), and that AdaGrad and AdaDelta, in their optimal hyperparameter settings, provide very similar loss values.

Since AdaGrad has only one hyperparameter η and a lower complexity (it only requires one per parameter accumulator and a rescaling operation

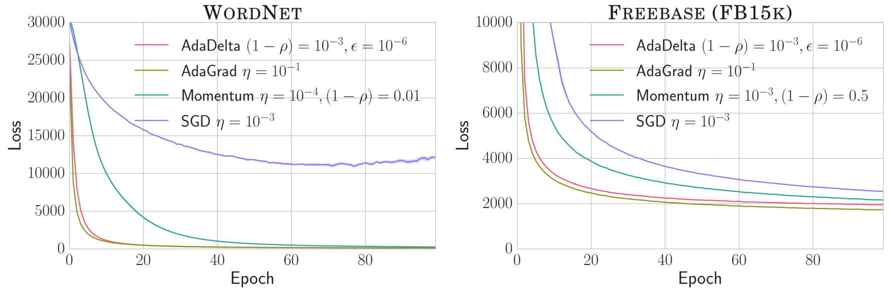

Fig. 3: Average loss across 10 TransE parameters learning tasks on the WordNet (left) and Freebase FB15k (right) knowledge graphs, using the optimal settings in [6]. For each of the optimization methods, the hyperparameters settings that after 100 epochs achieve the lowest average loss are reported.

at each iteration) than AdaDelta, we select AdaGrad (with $\eta = 0.1$) as the optimization method of choice. Specifically, as a successive step, we needed to assess whether AdaGrad (with $\eta = 0.1$) leads to *more accurate models*, i.e. with lower Mean Rank and higher Hits@10, than SGD. For the purpose, we trained TransE by using AdaGrad (with $\eta = 0.1$) for 100 epochs on a link prediction task on the WordNet and Freebase (FB15k) knowledge graphs, under the same evaluation setting used in [6]. Hyperparameters were selected according to the performance on the validation set using the same grid of hyperparameters adopted in [6]. Specifically, we chose the margin $\gamma \in \{1, 2, 10\}$, the embedding vector dimension $k \in \{20, 50\}$, and the dissimilarity $d \in \{L_1, L_2\}$. Tab. 2 shows the results obtained by TransE trained using AdaGrad (with $\eta = 0.1$) for 100 epochs, in comparison with state-of-the-art results as reported in [6]. From the table it is possible to see that, despite of

Table 2: **Link Prediction Results:** Test performance of several state-of-the-art Link Prediction methods on the WordNet and Freebase (FB15k) KGs. Results show the Mean Rank (the lower, the better) and Hits@10 (the higher, the better) for both the Raw and the Filtered settings [6].

Knowledge Graph	WordNet				Freebase (FB15k)			
Metric	Mean Rank		Hits@10 (%)		Mean Rank		Hits@10 (%)	
	Raw	Filt.	Raw	Filt.	Raw	Filt.	Raw	Filt.
Unstructured [5]	315	304	35.3	38.2	1074	979	4.5	6.3
RESCAL [21]	1180	1163	37.2	52.8	828	683	28.4	44.1
SE [7]	1011	985	68.5	80.5	273	162	28.8	39.8
SME linear [5]	545	533	65.1	74.1	274	154	30.7	40.8
SME bilinear [5]	526	509	54.7	61.3	284	158	31.3	41.3
LFM [14]	469	456	71.4	81.6	283	164	26.0	33.1
TransE [6]	263	251	75.4	89.2	243	125	34.9	47.1
TransE (AdaGrad)	**169**	**158**	**80.5**	**93.5**	**189**	**73**	**44.0**	**60.1**

the sensibly lower number of training epochs (100, compared to 1000 used for training TransE with SGD, as reported by [6]), TransE trained using AdaGrad provides more accurate link prediction models (i.e. lower Mean Rank and higher Hits@10 values) than every other model in the comparison.

The results showed in this section largely prove that our proposed solution is able to give a positive answer to **Q1**. Specifically, besides of experimentally proving that the adaptive learning rates proposed in Sect. 3.2 are able to improve the efficiency of parameters learning with respect to the current state-of-the-art EBEMs, we have also proved that the final learned model is able to outperform current state-of-the-art models in terms of Mean Rank and Hits@10.

5.2 Evaluation of the Proposed Energy Functions

In this section, we evaluate the energy functions proposed in Sect. 3.1 in the definition of an EBEM, with the final goal of providing reply to question **Q2**, that is to assess whether the energy functions proposed in Sect. 3.1 lead to more accurate link prediction models for KGs completion with respect to the state-of-the-art.

As from (1), the energy function of an EBEM can be rewritten as:

$$E(\langle s, p, o \rangle) = g(f_s(\mathbf{e}_s, \mathbf{S}_p), f_o(\mathbf{e}_o, \mathbf{S}_p))$$

where $\mathbf{e}_s$ and $\mathbf{e}_o$ denote the embedding vectors of the subject s and the object o of the triple, and $\mathbf{S}_p$ denotes the set of embedding parameters associated to the predicate p.

In Sect. 3.1 we proposed alternative choices for functions $f_s(\cdot)$ and $f_o(\cdot)$, that allow defining models whose number of parameters grows *linearly* with the number of entities and relations in the KG. Specifically, we proposed using *translation*, *scaling*, their composition, and the projection on the Euclidean unit sphere $n(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|$.

For each of the considered choices, we trained the corresponding EBEM on the WordNet and the Freebase (FB15k) knowledge graphs. Hyperparameters were selected on the basis of the model performances on the validation set: we selected the embedding vector dimension $k \in \{20, 50, 100\}$, the margin $\gamma \in \{2, 5, 10\}$, and the $g(\cdot)$ function $g(\mathbf{x}, \mathbf{y}) \in \{\|\mathbf{x}-\mathbf{y}\|_1, \|\mathbf{x}-\mathbf{y}\|_2, -\mathbf{x}^T\mathbf{y}\}$, corresponding to the L_1 and L_2 distances, and the negative dot product. Following the results from Sect. 5.1, model parameters were learned using AdaGrad (with $\eta = 0.1$) for 100 training epochs.

Tab. 3 shows the test results obtained with different choices of $f_s(\cdot)$ and $f_o(\cdot)$ functions. Additionally, for the purpose of comparison, we also add the results obtained by TransE (as reported in [6]) standing for the best performing model in the literature, on the same link prediction tasks.

From the table, it is interesting to note that, especially for highly multi-relational KGs such as Freebase (FB15k), *simpler models* for $f_s(\cdot)$ and $f_o(\cdot)$ provide better results than their more complex variants. A possible motivation for this is that a number of relations in FB15k only occur in a limited number of triples (only 736 predicates out of 1345 occur in more than 20 triples) and in cases like this more expressive models are less able to generalize correctly that simpler models. Given $f_o(\mathbf{e}_o, \{\mathbf{e}_p\}) = \mathbf{e}_o$, the best performing models, in terms of Hits@10, are:

- $f_s(\mathbf{e}_s, \{\mathbf{e}_p\}) = \mathbf{e}_s + \mathbf{e}_p$, representing the predicate-dependent *translation* of the subject's embedding vector
- $f_s(\mathbf{e}_s, \{\mathbf{e}_p\}) = \mathbf{e}_s \odot \mathbf{e}_p$, representing the predicate-dependent *scaling*.

This indicates that, despite the very different geometric interpretations, relying on simpler models improves link prediction results, especially in highly-relational knowledge graphs.

We can conclude that, constraining the expressiveness of the models while using adaptive learning rates, yields a significant improvement over state-of-the-art methods discussed in [6].

Table 3: **Link Prediction Results:** Test performance of several EBEMs (corresponding to different choices of the $f_s(\cdot)$ and $f_o(\cdot)$ functions) in comparison with TransE [6] on the WordNet and Freebase (FB15k) knowledge graphs. Results show the Mean Rank (the lower, the better) and Hits@10 (the higher, the better) in the Raw and Filtered settings.

Knowledge Graph	WordNet				Freebase (FB15k)			
Metric	Mean Rank		Hits@10 (%)		Mean Rank		Hits@10 (%)	
	Raw	Filt.	Raw	Filt.	Raw	Filt.	Raw	Filt.
TransE [6]	263	251	75.4	89.2	243	125	34.9	47.1
$f_s = \mathbf{e}_s + \mathbf{e}_p$ $f_o = \mathbf{e}_o$	**161**	**150**	**80.5**	**93.5**	**189**	**65**	**47.9**	**67.6**
$f_s = \mathbf{e}_s \odot \mathbf{e}_p$ $f_o = \mathbf{e}_o$	**229**	**215**	**81.4**	**93.5**	**207**	**81**	**46.5**	**65.3**
$f_s = (\mathbf{e}_s \odot \mathbf{e}_{p,1}) + \mathbf{e}_{p,2}$ $f_o = \mathbf{e}_o$	168	155	81.3	93.2	214	88	41.8	57.3
$f_s = \mathbf{e}_s + \mathbf{e}_{p,1}$ $f_o = \mathbf{e}_o + \mathbf{e}_{p,2}$	171	159	79.6	92.6	196	78	44.9	62.4
$f_s = \mathbf{e}_s \odot \mathbf{e}_{p,1}$ $f_o = \mathbf{e}_o \odot \mathbf{e}_{p,2}$	337	325	**83.0**	**95.2**	202	75	44.9	62.9
$f_s = (\mathbf{e}_s \odot \mathbf{e}_{p,1}) + \mathbf{e}_{p,2}$ $f_o = \mathbf{e}_o \odot \mathbf{e}_{p,3}$	279	266	82.4	94.3	210	88	42.3	59.1
$f_s = (\mathbf{e}_s \odot \mathbf{e}_{p,1}) + \mathbf{e}_{p,2}$ $f_o = (\mathbf{e}_o \odot \mathbf{e}_{p,3}) + \mathbf{e}_{p,4}$	320	308	81.6	93.6	211	87	40.0	54.9
$f_s = n(\mathbf{e}_s + \mathbf{e}_p)$ $f_o = \mathbf{e}_o$	211	200	75.7	88.7	237	115	39.5	55.4
$f_s = n(\mathbf{e}_s \odot \mathbf{e}_p)$ $f_o = \mathbf{e}_o$	226	213	77.6	89.2	262	132	42.0	59.9
$f_s = n((\mathbf{e}_s \odot \mathbf{e}_{p,1}) + \mathbf{e}_{p,2})$ $f_o = \mathbf{e}_o$	160	148	77.7	88.7	239	103	42.8	59.1
$f_s = n(\mathbf{e}_s + \mathbf{e}_{p,1})$ $f_o = n(\mathbf{e}_o + \mathbf{e}_{p,2})$	262	251	79.3	91.6	206	86	**47.5**	**66.5**
$f_s = n(\mathbf{e}_s \odot \mathbf{e}_{p,1})$ $f_o = n(\mathbf{e}_o \odot \mathbf{e}_{p,2})$	761	750	73.4	83.5	249	120	42.0	61.0
$f_s = n(\mathbf{e}_s \odot \mathbf{e}_{p,1} + \mathbf{e}_{p,2})$ $f_o = n(\mathbf{e}_o \odot \mathbf{e}_{p,3} + \mathbf{e}_{p,4})$	624	613	74.7	83.6	238	114	42.7	60.4

Source code and datasets for reproducing the experiments presented in this paper are available on-line[7].

6 Conclusions and Future Works

We focused on Energy-Based Embedding Models, a novel class of link prediction models for knowledge graph completion where each entity in the graph is represented by a continuous *embedding* vector.

Models in this class, like the *Translating Embedding* model [6], have been used to achieve performances that are comparable with the main state-of-the-art methods while scaling on very large knowledge graphs.

In this work, we proposed: (i) a general framework for describing state-of-the-art Energy-Based Embedding Models, (ii) a family of novel energy functions, with useful properties, (iii) a method for improving the efficiency of the learning process by an order of magnitude, while leading to more accurate link prediction models.

We empirically evaluated the adoption of the proposed adaptive learning rates in the context of Energy-Based Embedding Models by showing that they provide more accurate link prediction models while reducing the learning time by an order of magnitude in comparison with state-of-the-art learning algorithms. We also empirically evaluated the newly proposed energy functions (with a number of parameters) that scales *linearly* with the number of entities and relations in the knowledge graph. Our results showed a significant improvement over state-of-the-art link prediction methods on the very same considered large knowledge graphs, that are WordNet and Freebase.

For the future we plan to investigate on the formalization of Energy-Based Embedding Models that are able to take into account the available background knowledge. Other research directions include dynamically controlling the complexity of learned models, and further optimizing the learning process.

References

1. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. Journal of Machine Learning Research 9, 1981–2014 (2008)
2. Bartlett, P.L., et al. (eds.): Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States (2012)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - A crystallization point for the web of data. J. Web Sem. 7(3), 154–165 (2009)
4. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Wang, J.T. (ed.) Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008. pp. 1247–1250. ACM (2008)

[7] `https://github.com/pminervini/ebemkg/`

5. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. Machine Learning 94(2), 233–259 (2014)
6. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Burges et al. [8], pp. 2787–2795
7. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Burgard, W., et al. (eds.) Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011. AAAI Press (2011)
8. Burges, C.J.C., et al. (eds.): Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States (2013)
9. Chang, K., Yih, W., Yang, B., Meek, C.: Typed tensor decomposition of knowledge bases for relation extraction. In: Moschitti, A., et al. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1568–1579. ACL (2014)
10. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Le, Q.V., Mao, M.Z., Ranzato, M., Senior, A.W., Tucker, P.A., Yang, K., Ng, A.Y.: Large scale distributed deep networks. In: Bartlett et al. [2], pp. 1232–1240
11. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Macskassy, S.A., et al. (eds.) The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014. pp. 601–610. ACM (2014)
12. Duchi, J.C., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12, 2121–2159 (2011)
13. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning. The MIT Press (2007)
14. Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. In: Bartlett et al. [2], pp. 3176–3184
15. Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., Ueda, N.: Learning systems of concepts with an infinite relational model. In: Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA. pp. 381–388. AAAI Press (2006)
16. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
17. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. In: Bakir, G., et al. (eds.) Predicting Structured Data. MIT Press (2006)
18. Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: A knowledge base from multilingual wikipedias. In: CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Online Proceedings (2015)
19. Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM 38(11), 39–41 (1995)
20. Miller, K.T., Griffiths, T.L., Jordan, M.I.: Nonparametric latent feature models for link prediction. In: Bengio, Y., et al. (eds.) Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada. pp. 1276–1284. Curran Associates, Inc. (2009)
21. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Getoor, L., et al. (eds.) Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011. pp. 809–816. Omnipress (2011)
22. Rettinger, A., Nickles, M., Tresp, V.: Statistical relational learning with formal ontologies. In: Buntine, W.L., et al. (eds.) Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II. LNCS, vol. 5782, pp. 286–301. Springer (2009)

23. Rumelhart, D.E., Hinton, G.E., Wilson, R.J.: Learning representations by back-propagating errors. Nature 323, 533–536 (1986)
24. Schaul, T., Antonoglou, I., Silver, D.: Unit tests for stochastic optimization. In: International Conference on Learning Representations. Banff, Canada (2014)
25. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Burges et al. [8], pp. 926–934
26. Wang, Y.J., Wong, G.Y.: Stochastic blockmodels for directed graphs. Journal of the American Statistical Association 82(397), pp. 8–19 (1987)
27. Xu, Z., Tresp, V., Yu, K., Kriegel, H.: Infinite hidden relational models. In: UAI '06, Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence, Cambridge, MA, USA, July 13-16, 2006. AUAI Press (2006)
28. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR abs/1212.5701 (2012)