Leveraging the Schema in Latent Factor Models for Knowledge Graph Completion

Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, Floriana Esposito
Dipartimento di Informatica, LACAM Laboratory
Università degli Studi di Bari Aldo Moro
Via E. Orabona, 4 - 70125 Bari - Italy
{ pasquale.minervini, claudia.damato, nicola.fanizzi, floriana.esposito } @uniba.it

ABSTRACT

We focus on the problem of predicting missing links in large Knowledge Graphs (KGs), so to discover new facts. Over the last years, *latent factor models* for link prediction have been receiving an increasing interest: they achieve state-of-the-art accuracy in link prediction tasks, while scaling to very large KGs. However, KGs are often endowed with additional *schema knowledge*, describing entity classes, their sub-class relationships, and the domain and range of each predicate: the schema is actually not used by latent factor models proposed in the literature. In this work, we propose an unified method for leveraging additional schema knowledge in latent factor models, with the aim of learning more accurate link prediction models. Our experimental evaluations show the effectiveness of the proposed method on several KGs.

CCS Concepts

•**Computing methodologies → Semantic networks; Reasoning about belief and knowledge;**

Keywords

Link Prediction, Knowledge Graph, Latent Factor Model

1. INTRODUCTION

Knowledge Graphs (KGs) are graph-structured knowledge bases, where factual knowledge is represented in the form of relationships between entities: they are a powerful instrument for search, analytics, recommendations, and data integration. Several KGs are publicly available through the *Linked Open Data* (LOD) cloud, a collection of interlinked KGs such as Freebase [2], DBpedia [1] and YAGO [10]. 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 [1]. However, KGs are often largely incomplete. For instance consider Freebase [2], a core element in the Google Knowledge Vault project [6]: 71% of the persons described in Freebase have no known place of birth, 75% of them have no known nationality, while the coverage for less frequent predicates can be even lower [6].

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*. For solving this problem, over the last years, *latent factor models* have been receiving an increasing interest [3]. In particular, the recently proposed *Translating Embeddings* model [5] was shown to achieve state-of-the-art link prediction results, while being able to scale to very large and highly-relational KGs [3].

In KGs, facts are represented by $\langle s, p, o \rangle$ triples, where each triple denotes a relationship of type p (predicate of the triple) between the subject s and the object o. Latent factor models associate a *prediction score* to each triple, measured as a function of the *latent factors* (also referred to as *latent features* [11]) associated to the subject, the predicate and the object of the triple. The latent factors of all entities and predicates in the KG are learned jointly, by maximizing the compatibility between the latent factors and the KG. As a result, the learned latent factors retain global, structural information about the KG [11]. However, real world KGs are usually endowed with additional ontological *schema knowledge* containing high level knowledge about the KG. Latent factor models proposed in the literature are not designed for leveraging the schema of a KG: how to correctly include schema knowledge in such models is a largely unexplored field. In this work, we propose a principled, unified method for including schema knowledge in latent factor models, with the aim of learning more accurate link prediction models.

This paper is structured as follows. In Sect. 2, we introduce the basic concepts. In Sect. 3 we analyze several state-of-the-art models, and discuss their limitations. In Sect. 4 we propose a novel method for leveraging schema knowledge in latent factor models for link prediction. In Sect. 5 we extend the learning process so to learn a set of new schema-related parameters. In Sect. 6 we experimental show the effectiveness of the proposed method on several datasets. In Sect. 7 we summarize this work and discuss future research directions.

SAC 2016,April 04-08, 2016, Pisa, Italy
Copyright 2016 ACM 978-1-4503-3739-7/16/04…$15.00
http://dx.doi.org/10.1145/2851613.2851841

[1] State of the LOD Cloud 2014: http://lod-cloud.net/
[2] Available at https://developers.google.com/freebase/data

2. BASICS

RDF Graphs - In this work, we assume the KG is represented using the W3C *Resource Description Framework* (RDF) [3], a recommended standard for expressing information about resources. Resources can be anything, including documents, people, physical objects, and abstract concepts.

An RDF knowledge base, also called *RDF graph*, 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 subject s and the object o denote *resources*, or *entities*, and p denotes a *predicate*, or *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.*

Example 2.1 (Shakespeare). *Consider the following statement:* William Shakespeare, author of the tragedy Hamlet and the Othello, was influenced by Geoffrey Chaucer*. It can be expressed by the following RDF triples:*

$$\begin{array}{rrr} \langle \mathit{Shakespeare}, & \mathit{authorOf}, & \mathit{Hamlet}\rangle \\ \langle \mathit{Hamlet}, & \mathit{genre}, & \mathit{Tragedy}\rangle \\ \langle \mathit{Shakespeare}, & \mathit{authorOf}, & \mathit{Othello}\rangle \\ \langle \mathit{Shakespeare}, & \mathit{influencedBy}, & \mathit{Chaucer}\rangle \end{array}$$

An RDF graph intrinsically represents a *labeled directed multigraph*, where each entity is a vertex, and each RDF triple is represented by an edge whose label is a predicate and emanating from its subject vertex to its object vertex. In RDF, the *Open World Assumption* holds: a missing triple does not mean that the corresponding statement is false, but rather that its truth value is *missing*, meaning that it cannot be observed in the KG.

In the following, given an RDF graph G, we denote by $\mathcal{E}_G = \{s \mid \exists\langle s, p, o\rangle \in G\} \cup \{o \mid \exists\langle s, p, o\rangle \in G\}$ the set of all entities occurring in G, and by $\mathcal{R}_G = \{p \mid \exists\langle s, p, o\rangle \in G\}$ the set of all predicates occurring in G. For instance, in the case of the RDF graph shown in Ex. 2.1, we have that $\mathcal{E}_G = \{\texttt{Shakespeare}, \texttt{Hamlet}, \texttt{Tragedy}, \texttt{Othello}, \texttt{Chaucer}\}$, and $\mathcal{R}_G = \{\texttt{authorOf}, \texttt{genre}, \texttt{influencedBy}\}$. We will denote by $\mathcal{S}_G = \mathcal{E}_G \times \mathcal{R}_G \times \mathcal{E}_G$ the set of *possible triples* that can be generated using entities and predicates in G. We refer to all triples in G as *visible triples*, and to all triples in $\mathcal{S}_G \setminus G$ as *unobserved triples.*

Due to the Open World Assumption, unobserved triples might encode true statements. For instance, consider the triple $\langle \texttt{Othello}, \texttt{genre}, \texttt{Tragedy}\rangle$: although it is unobserved, it represents the true statement *Othello is a Tragedy.* In this work, we focus on identifying unobserved triples representing true statements, so to provide a likely completion of the KG.

RDF Schema - RDF Schema (RDFS) [4]extends the RDF vocabulary by providing mechanisms for describing groups of related entities and the relationships between these entities.

Let the prefixes `rdf` and `rdfs` represent the RDF and RDFS namespaces, respectively. RDF provides a property between resources, `rdf:type`, that relates a resource to the types that the resource belongs to. RDFS extends the RDF vocabulary by defining several built-in classes, such as the class of all classes `rdfs:Class`, the class of all properties `rdfs:Property`, and the class of all resources `rdfs:Resources`. RDFS also defines several relationships between classes (such as the subclass relationship `rdfs:subClassOf`) and between properties and classes (such as the domain relationship `rdfs:domain`, and the and range relationship `rdfs:range`).

Example 2.2 (Shakespeare – cont.). *The RDF graph in Ex. 2.1 can be enriched with RDFS schema knowledge. For instance, consider the following statement:* William Shakespeare is a person, and the authorship relation can only occur between a person and a literary work*. It can be expressed by the following RDF triples, through the RDF(S) vocabularies:*

$$\begin{array}{rrr} \langle \mathit{Shakespeare}, & \mathit{rdf{:}type}, & \mathit{Person}\rangle \\ \langle \mathit{authorOf}, & \mathit{rdfs{:}domain}, & \mathit{Person}\rangle \\ \langle \mathit{authorOf}, & \mathit{rdfs{:}range}, & \mathit{LiteraryWork}\rangle \end{array}$$

The RDFS *entailment regime* [5] defines a set of *logical entailment rules* which allow to deductively infer new and correct RDF statements from a given RDF graph. For instance, the following two RDFS entailment rules:

$$\begin{array}{rl} \{\langle p, \texttt{rdfs:domain}, c\rangle, \langle s, p, o\rangle\} \Rightarrow & \langle s, \texttt{rdf:type}, c\rangle, \\ \{\langle p, \texttt{rdfs:range}, c\rangle, \langle s, p, o\rangle\} \Rightarrow & \langle o, \texttt{rdf:type}, c\rangle, \end{array}$$

can be interpreted as follows: *If the predicate p has domain (resp. range) c, and an entity occurs as a subject (resp. object) of p, then such entity belongs to the class c.*

Latent Factor Models - Latent factor models for link prediction represent the confidence in each triple (fact) as a function of a set of *latent factors* (also called *latent features*) associated to the subject, predicate and object of the triple; see [11] for a recent overview. The term *latent* refers to the fact that such features are not directly observable in the KG, but rather they are *estimated* from data. For example, a possible explanation of the influence of William Shakespeare and his plays in a large number of contemporary works of art, such as writings and movies (observable evidence), is that Shakespeare was a great writer (latent feature).

More specifically, a latent factor model associates a *prediction score* $\theta_{s,p,o} = f(\langle s, p, o\rangle; \Theta)$ to each triple $\langle s, p, o\rangle$, where $f(\,\cdot\,; \Theta)$ is a model-dependent scoring function, and Θ are the model parameters. The score $\theta_{s,p,o}$ represents the model's confidence that the statement represented by $\langle s, p, o\rangle$ holds true, while Θ represents the latent factors used by the model for *explaining* the evidence [11]. In a *link prediction* setting, given the model parameters Θ, the scoring function $f(\,\cdot\,; \Theta)$ is used for ranking unobserved triples in $\mathcal{S}_G \setminus G$: those with a higher prediction score have a higher probability of representing a true statement, and are considered for a completion of the KG G. The model parameters Θ can be *learned from data*: this aspect is discussed in detail in Sect. 5.

In the following sections, we briefly survey the models proposed in the literature: we analyze their limitations, and propose a principled method for leveraging RDFS schema knowledge in latent factor models, with the aim of learning more accurate link prediction models.

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

[4] http://www.w3.org/TR/rdf-schema/

[5] http://www.w3.org/TR/rdf11-mt/

3. LATENT FACTOR MODELS FOR LINK PREDICTION

Several latent factor models have been proposed in the literature for solving the link prediction problem. Some widely cited models are RESCAL [12], the *Semantic Matching Energy* model [4] and the *Translating Embeddings* model [5]. Such models differ by the choice of the scoring function $f(\,\cdot\,;\Theta)$, where Θ denotes the set of model parameters, and how such parameters are learned from the knowledge graph.

A common characteristic of these models is that they associate a unique *latent factor* $\mathbf{e}_x \in \mathbb{R}^k$ to each entity $x \in \mathcal{E}_G$ in the KG, where $k \in \mathbb{N}$ is a user-defined hyper-parameter. Each vector $\mathbf{e}_x \in \mathbb{R}^k$ can be interpreted as a collection of *latent features* describing the entity x [11]. Given a triple $\langle s, p, o \rangle$, its prediction score $\theta_{s,p,o} = f(\langle s, p, o \rangle; \Theta)$ is calculated as a function of the latent factors $\mathbf{e}_s$ and $\mathbf{e}_o$, respectively associated to the subject s and the object o of the triple.

The *Translating Embeddings* model, also referred to as TransE, proposed in [4] is particularly interesting: despite its simple formulation, it represents the state-of-the-art in latent factor models for link prediction in knowledge graphs in terms of predictive accuracy. Furthermore, it overcomes many of the limitations in terms of efficiency and scalability of related models proposed in the literature, and it was shown to scale to very large and highly-relational knowledge graphs [3].

The Translating Embeddings Model.

In TransE, each entity and predicate $x \in \mathcal{E}_G \cup \mathcal{R}_G$ is mapped to a unique, low-dimensional latent factor $\mathbf{e}_x \in \mathbb{R}^k$, also referred to as the *embedding vector* of x. Given a triple $\langle s, p, o \rangle$, the corresponding prediction score $\theta_{s,p,o}$ is given by:

$$\theta_{s,p,o} = f(\langle s, p, o \rangle; \Theta) = -\delta(\mathbf{e}_s + \mathbf{e}_p, \mathbf{e}_o), \qquad (1)$$

where $\Theta = \{\mathbf{e}_x \in \mathbb{R}^k \mid x \in \mathcal{E}_G \cup \mathcal{R}_G\}$ is the set of model parameters, corresponding to the set of all embedding vectors, and $\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{R}^k$ are the embedding vectors associated to the subject s, the predicate p, and the object o, respectively. The function $\delta(\,\cdot\,)$ in Eq. 1 is a *dissimilarity function* corresponding to either the L_1 or the L_2 distance, i.e. $\delta(\mathbf{x}, \mathbf{y}) \in \{\|\mathbf{x} - \mathbf{y}\|_1, \|\mathbf{x} - \mathbf{y}\|_2\}$.

Let $N_e = |\mathcal{E}_G|$ and $N_r = |\mathcal{R}_G|$ denote the number of entities and predicates in the KG. In TransE, the number of model parameters is $|\Theta| = N_e k + N_r k$, a quantity that grows *linearly* with N_e and N_r, and with the latent factor dimension k. Furthermore, evaluating the scoring function in Eq. 1 and its gradient w.r.t. the model parameters only requires simple, element-wise vector operations. For such a reason, TransE can scale to large and highly-relational KGs [4].

Despite its simplicity, TransE yields more accurate link prediction results than more complex models in the literature, such as RESCAL [3]. A possible explanation is that, in TransE, a lower number of model parameters, together with a simple linear interaction between the latent factors, lead to better generalization properties [5].

The Scaling Embeddings Model.

In TransE, each prediction score $\theta_{s,p,o} = f(\langle s, p, o \rangle; \Theta)$ is based on an *additive interaction* of the latent factors $\mathbf{e}_s$ and $\mathbf{e}_p$ associated to the subject and the predicate of the triple. In a recent work [13], authors found that a *multiplicative interaction* of the latent factors can be a better alternative, as it scales each component in $\mathbf{e}_s$ with the strength of the corresponding component in $\mathbf{e}_p$. For such a reason, during experiments, we also consider the following TransE variants:

$$\begin{aligned}
&\mathsf{TransE}^+ && : f(\langle s, p, o \rangle; \Theta) = -\delta(\mathbf{e}_s + \mathbf{e}_{p,1}, \mathbf{e}_o + \mathbf{e}_{p,2}),\\
&\mathsf{ScalE} && : f(\langle s, p, o \rangle; \Theta) = -\delta(\mathbf{e}_s \circ \mathbf{e}_p, \mathbf{e}_o),\\
&\mathsf{ScalE}^+ && : f(\langle s, p, o \rangle; \Theta) = -\delta(\mathbf{e}_s \circ \mathbf{e}_{p,1}, \mathbf{e}_o \circ \mathbf{e}_{p,2}),
\end{aligned}$$

where $\circ$ denotes the element-wise product, corresponding to the vector *scaling* operation, and $\delta(\mathbf{x}, \mathbf{y})$ is a dissimilarity function corresponding either to the L_1 or L_2 distance (as in the original TransE model), or to the negative inner product, i.e. $\delta(\mathbf{x}, \mathbf{y}) \in \{\|\mathbf{x} - \mathbf{y}\|_1, \|\mathbf{x} - \mathbf{y}\|_2, -\mathbf{x}^T\mathbf{y}\}$.

The model denoted by ScalE applies a *scaling operation*, instead of a translation, to the latent factor $\mathbf{e}_s$ associated to the subject of the triple s. TransE$^+$ and ScalE$^+$ increase the expressiveness of TransE and ScalE by associating *two* latent factors to each predicate, and translating (resp. scaling) the latent factors associated to both the subject and the object.

However, despite their widespread use in link prediction tasks [3], latent factor models proposed in literature are not yet capable of leveraging the schema knowledge, encoded in RDFS, available for many KGs. In the following section, we address this problem by proposing an unified method for including schema knowledge in latent factor models, with the aim of learning more accurate link prediction models.

4. LEVERAGING SCHEMA KNOWLEDGE

The vast majority of KGs, including Freebase [2], DBpedia [1] and YAGO [10], are endowed with additional RDFS *schema knowledge*, which is not taken into account by latent factor models proposed in the literature. In this section, we propose an unified method for leveraging schema knowledge in latent factor models, with the aim of learning more accurate prediction models. Consider the following example:

EXAMPLE 4.1 (SHAKESPEARE – CONT.). *The RDF(S) graph in Ex. 2.2 can be further enriched with schema knowledge representing the following statement:* Othello is a literary work, England is a location, and the domain of the genre relation is literary works. *The statement can be represented by the following RDF triples:*

$$\begin{aligned}
&\langle \mathtt{Othello}, && \mathtt{rdf{:}type}, && \mathtt{LiteraryWork}\rangle\\
&\langle \mathtt{England}, && \mathtt{rdf{:}type}, && \mathtt{Location}\rangle\\
&\langle \mathtt{genre}, && \mathtt{rdfs{:}domain}, && \mathtt{LiteraryWork}\rangle
\end{aligned}$$

Now, consider the problem of scoring two triples T_1 (Othello is a Tragedy) *and* T_2 (England is a Tragedy), *according to the confidence that the corresponding statement holds true:*

$$\begin{aligned}
T_1 : \quad &\langle \mathtt{Othello}, && \mathtt{genre}, && \mathtt{Tragedy}\rangle\\
T_2 : \quad &\langle \mathtt{England}, && \mathtt{genre}, && \mathtt{Tragedy}\rangle
\end{aligned}$$

We can add either T_1 *or* T_2 *to the KG, without causing any inconsistency. However, adding* T_2 *to the KG causes the resource* `England` *to be additionally typed as a* `LiteraryWork`*,*

Table 1: Statistics for the datasets used in link prediction experiments

Dataset	Entities	Predicates	Training Triples	Validation Triples	Test Triples
Freebase (FB15k)	14,951	1,345	483,142	50,000	59,071
YAGO3	120,004	35	1,082,107	1,000	1,000
DBpedia 2014 (Music)	104,422	7	265,156	1,000	1,000

according to the RDFS entailment rules. This is still correct, since RDFS does not allow expressing that two classes are disjoint, but it may denote a potential modeling flaw [8]. For instance, in this particular case, England is meant as a location, and clearly not as a literary work.

In a *link prediction* setting, we aim at predicting missing facts (triples) in a knowledge graph. In this context, Othello can be considered as *more likely* to appear as a literary work of the Tragedy genre than England.

We propose a principled, unified method for leveraging RDFS schema information in latent factor models. Specifically, we introduce a set of predicate-specific parameters $\boldsymbol{\lambda}$ that *adaptively decrease* the prediction score of triples if they imply, according to the RDFS logical entailment rules, previously unknown and possibly conflicting type information on the subject or object of such triples. The proposed method allows assigning lower prediction scores to unobserved triples that, once added to the KG, introduce unlikely type information. The introduced schema-related parameters $\boldsymbol{\lambda}$ can be learned jointly with the model parameters Θ.

Formally, for each predicate $p \in \mathcal{R}_G$, let $\text{domain}_G(p) \subseteq \mathcal{E}_G$ denote the set of entities typed as the domain of p. Similarly, let $\text{range}_G(p) \subseteq \mathcal{E}_G$ denote the set of entities typed as the range of p, according to RDFS entailment rules. We introduce two *penalty terms* $g_G : \mathcal{R}_G \times \mathcal{E}_G \to \{0, 1\}$ and $h_G : \mathcal{R}_G \times \mathcal{E}_G \to \{0, 1\}$, defined as follows:

$$
\begin{aligned}
g_G(p, s) &= \begin{cases} 1 & \text{if } s \notin \text{domain}_G(p), \\ 0 & \text{otherwise.} \end{cases} \\
h_G(p, o) &= \begin{cases} 1 & \text{if } o \notin \text{range}_G(p), \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
\tag{2}
$$

Given the scoring function $f(\,\cdot\,; \Theta)$ associated to a given latent factor model, we define the corresponding *schema-aware scoring function* $f_{\text{S}}(\,\cdot\,; \Theta, \boldsymbol{\lambda})$ as follows:

$$
\begin{aligned}
f_{\text{S}}(\langle s, p, o \rangle; \Theta, \boldsymbol{\lambda}) = &f(\langle s, p, o \rangle; \Theta) \\
&- \lambda_p^g g_G(p, s) - \lambda_p^h h_G(p, o),
\end{aligned}
\tag{3}
$$

where $\boldsymbol{\lambda} = \{\lambda_p^g, \lambda_p^h \in \mathbb{R}^+ \mid p \in \mathcal{R}_G\}$ is a set of additional schema-related parameters: in particular, λ_p^g and λ_p^h are two new predicate-specific parameters, associated with the predicate $p \in \mathcal{R}_G$, that control the weight of the penalty terms g_G and h_G. The new schema-aware model relies on $|\boldsymbol{\lambda}| = 2N_r$ additional schema-related parameters, a quantity that scales *linearly* with the number of predicates in the KG.

The role of parameters λ_p^g and λ_p^h in weighting the penalty terms is the following. Setting $\lambda_p^g = 0$ and $\lambda_p^h = 0$ corresponds to not leveraging any additional schema knowledge, since it follows that $f_{\text{S}}(\langle s, p, o \rangle; \Theta, \boldsymbol{\lambda}) = f(\langle s, p, o \rangle; \Theta)$.

When $\lambda_p^g > 0$, we are associating a higher prediction score to triples $\langle s, p, o \rangle$ where, given a predicate p (e.g. genre), the subject s (e.g. Othello) is typed as the domain of p (e.g. Literary Work). Similarly, when $\lambda_p^h > 0$, we are associating a higher prediction score to triples where the object o (e.g. Tragedy) is typed as the range of p (e.g. Literary Genre).

In Ex. 4.1, assume that the scoring function of a latent factor model $f(\,\cdot\,; \Theta)$, for some reason (such as lack of statistical evidence) assigns the same prediction score to T_1 and T_2, i.e. $f(T_1; \Theta) = f(T_2; \Theta)$. If $\lambda_p^g > 0$, we have that $f_{\text{S}}(T_1; \Theta, \boldsymbol{\lambda}) > f_{\text{S}}(T_2; \Theta, \boldsymbol{\lambda})$, i.e. the schema-aware model considers the statement encoded by T_1 (Othello is a Tragedy) as more likely than the statement encoded by T_2 (England is a Tragedy).

For each predicate $p \in \mathcal{R}_G$, the weights λ_p^g and λ_p^h can be fixed in advance, for instance by encoding an expert's domain knowledge. As an alternative, such weights can be *learned from data* jointly with the model parameters Θ. We discuss this aspect in the following section.

5. LEARNING THE MODEL PARAMETERS

In TransE and related models [5, 4], the optimal parameters Θ are learned from data. Specifically, they are estimated by incrementally increasing the prediction score of visible triples in G, while decreasing the score of unobserved triples in $\mathcal{S}_G \setminus G$. During the learning process, unobserved triples are randomly generated by *corrupting* visible triples, by replacing either their subject or their object with another entity in G. More formally, given an observed triple $T \in G$, let $\mathcal{C}_G(T)$ denote the set of all corrupted triples obtained by replacing either the subject or the object in T with another entity:

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

The optimal model parameters $\hat{\Theta}$ can be learned by minimizing a *margin-based ranking loss*, by solving the following optimization problem:

$$
\begin{aligned}
\underset{\Theta}{\text{minimize}} \quad & \sum_{T \in G} \sum_{\tilde{T} \in \mathcal{C}_G(T)} \left[\gamma - f(T; \Theta) + f(\tilde{T}; \Theta) \right]_+ \\
\text{subject to} \quad & \forall x \in \mathcal{E}_G : \; \|\mathbf{e}_x\| = 1,
\end{aligned}
\tag{4}
$$

where $[x]_+ = \max\{0, x\}$, and $\gamma \geq 0$ is a hyper-parameter referred to as *margin*. The loss function in Eq. 4 enforces the prediction score in observed triples to be higher than the score of their corrupted variants, by a margin of at least γ. The norm constraints in the optimization problem prevent to trivially solve the problem by increasing the norm of latent factors [4]. We refer to [5, 4] for more informations on solving the minimization problem in Eq. 4 by using the *Stochastic Gradient Descent* (SGD) algorithm. In this work, we propose learning the additional schema-related parameters $\boldsymbol{\lambda}$ jointly with the model parameters Θ by extending the minimization

Table 2: Link Prediction Results: Test performance of several latent factor models (f) and their schema-aware extensions (f_{S}) on the Freebase (FB15k), YAGO3 **and** DBpedia 2014 (Music) **datasets. Results show the** Mean Rank **(the lower, the better) and** Hits@10 **(the higher, the better) in the** Raw **and** Filtered **settings.**

Metric	Mean Rank				Hits@10 (%)			
	Raw		Filtered		Raw		Filtered	
	f	f_{S}	f	f_{S}	f	f_{S}	f	f_{S}
Dataset	**Freebase (FB15k)**							
Unstructured	595	**196**	488	**89**	9.7	**41.1**	13.7	**55.0**
TransE	213	**196**	86	**70**	44.2	**45.4**	62.3	**64.1**
ScalE	201	**172**	84	**55**	45.7	**47.2**	65.7	**68.2**
TransE^{+}	218	**202**	91	**75**	41.8	**42.7**	57.8	**59.4**
ScalE^{+}	218	**195**	82	**61**	48.1	$\boxed{\mathbf{49.8}}$	69.2	$\boxed{\mathbf{71.1}}$
Dataset	**YAGO3**							
Unstructured	6,548	**2,619**	4,792	**863**	7.4	**18.2**	10.3	**23.6**
TransE	2,541	**2,372**	1,438	**1,127**	28.3	**29.0**	42.1	**42.6**
ScalE	4,200	**2,643**	2,447	**886**	30.6	$\boxed{\mathbf{31.0}}$	45.3	$\boxed{\mathbf{45.7}}$
TransE^{+}	2,433	**2,396**	1,446	**1,381**	27.0	**28.0**	39.1	**40.0**
ScalE^{+}	3,475	**2,629**	1,716	**869**	29.0	29.0	41.0	41.0
Dataset	**DBpedia 2014**							
Unstructured	1,509	**922**	1,331	**745**	26.9	**36.1**	32.9	**43.0**
TransE	1,121	1,121	994	994	41.6	**43.3**	50.5	**52.1**
ScalE	1,299	**1,143**	1,149	**962**	46.8	$\boxed{\mathbf{47.5}}$	57.4	$\boxed{\mathbf{58.5}}$
TransE^{+}	1,173	1,173	1,059	1,059	40.5	**41.6**	50.1	**51.3**
ScalE^{+}	1,173	**1,134**	1,012	**973**	45.0	**45.3**	55.7	**56.0**

problem in Eq. 4 as follows:

$$\begin{aligned}
\underset{\Theta,\boldsymbol{\lambda}}{\text{minimize}} \quad & \sum_{T \in G} \sum_{\tilde{T} \in \mathcal{C}_G(T)} \left[\gamma - f_{\mathrm{S}}(T; \Theta, \boldsymbol{\lambda}) + f_{\mathrm{S}}(\tilde{T}; \Theta, \boldsymbol{\lambda}) \right]_{+} \\
\text{subject to} \quad & \forall x \in \mathcal{E}_G : \ \|\mathbf{e}_x\| = 1 \\
& \forall p \in \mathcal{R}_G : \ \lambda_p^g, \lambda_p^h \geq 0,
\end{aligned} \tag{5}$$

where an additional constraint enforces the schema-related weights $\boldsymbol{\lambda}$ to be non-negative. We propose solving the minimization problem in Eq. 5 by using SGD. In [13], authors experimentally found that using AdaGrad [7] for adaptively selecting the learning rates in SGD yields sensibly better results than those reported in [5]. In our experiments, we also use AdaGrad for selecting the optimal learning rates.

6. EXPERIMENTS

This section is organized as follows. In Sect. 6.1, we describe the datasets and evaluation metrics used in experiments. In Sect. 6.2, we empirically evaluate the *schema-aware extensions* to latent factor models, as proposed in Sect. 4.

6.1 Experimental Settings

In the experiments, we followed the evaluation protocols adopted in [5]. We evaluate the schema-aware scoring functions proposed in Sect. 4 on three datasets: Freebase (FB15k), YAGO3 and DBpedia 2014 (Music). Each dataset is composed by a *training*, a *validation* and a *testing* set of triples, as summarized in Tab. 1, obtained by randomly partitioning the triples in the RDF graph. Freebase (FB15k) is a dataset published in [5], enriched with RDFS triples freely available from the project website [6]. YAGO3 [10] is a large knowledge graph automatically extracted from several sources: our dataset is composed by the facts stored in the `yagoFacts` component of YAGO3, enriched with RDFS triples freely available from the project website [7]. DBpedia 2014 (Music) is a large DBpedia 2014 fragment extracted following the indications in [9], enriched with RDFS triples available from the project website [8].

Link Prediction. We use the metrics used in [5] for evaluating the *rank*, according to the model, of each test triple. Specifically, for each test triple $\langle s, p, o \rangle$, its object o is replaced with every entity $\tilde{o} \in \mathcal{E}_G$ in the knowledge graph G in turn, generating a set of *corrupted* triples in the form $\langle s, p, \tilde{o} \rangle$. The prediction scores of corrupted triples are first computed by the model, then sorted in descending order, and used to compute the rank of the correct triple. This procedure is repeated by corrupting the subject. Aggregated over all test triples, this procedure yields to the following metrics: *averaged rank* (denoted by Mean Rank) and *proportion of ranks not larger than* 10 (denoted by Hits@10). This is referred to as the Raw setting. However, if corrupting a triple generates another triple that exists in the knowledge graph, ranking it before the original triple is not wrong. For such a reason, corrupted triples that exist in either the training, validation or test sets were removed before computing the rank of each triple. This is referred to as the Filtered setting. In both the Raw and Filtered settings, a lower Mean Rank is better, while a higher Hits@10 is better.

6.2 Evaluation of the Schema-Aware Models

In the following experiments, we aim at assessing whether the schema-aware extensions to latent factor models, proposed in Sect. 4, improve the predictive accuracy of such models in link prediction tasks. In particular, we compare TransE, ScalE, TransE^{+} and ScalE^{+} with their schema-aware variants. We also consider the Unstructured model, a latent factor model used in [4] and [5] as an informed baseline. Hyperparameters k, δ and γ were selected so to maximize the

[6] https://developers.google.com/freebase/data

[7] http://yago-knowledge.org

[8] http://downloads.dbpedia.org/2014/

model performance on the validation set: we selected k in $\{20, 50, 100, 200\}$, $\delta(\mathbf{x}, \mathbf{y})$ in $\{\|\mathbf{x} - \mathbf{y}\|_1, \|\mathbf{x} - \mathbf{y}\|_2, -\mathbf{x}^T\mathbf{y}\}$, and set $\gamma = 1$. For assessing whether the schema-aware extensions are beneficial to the predictive accuracy of latent factor models, in experiments we proceeded as follows. At first, we learned the parameters Θ of the scoring function $f(\,\cdot\,; \Theta)$, as described in Sect. 5. Then, we learned the schema-related parameters $\boldsymbol{\lambda}$, encoding the weights of penalty terms in the schema-aware scoring function $f_{\mathrm{S}}(\,\cdot\,; \Theta, \boldsymbol{\lambda})$, as proposed in Sect. 4. Results, summarized in Tab. 2, show the link prediction performance of models relying on the *classic* scoring function $f(\,\cdot\,; \Theta)$ (denoted by f), and its schema-aware variant $f_{\mathrm{S}}(\,\cdot\,; \Theta, \boldsymbol{\lambda})$ (denoted by f_{S}). As discussed in Sect. 5, model parameters Θ were learned by solving the optimization problem in Eq. 5 using SGD, and using AdaGrad [7] for selecting the optimal learning rates in SGD.

Results. In every experiment, we can see that the schema-aware extension of each latent factor model yields better results, in terms of the Hits@10 metric, in comparison with the original model. In particular, we can see that less accurate baseline Unstructured gains a huge benefit from the additional schema information: for instance, we can see that, on the Freebase dataset, its initial 13.7 Hits@10 improves to 55.0 Hits@10, becoming almost comparable with the 64.1 Hits@10 obtained with TransE. This is also true in the case of the YAGO3 and DBpedia 2014 dataset, where the Hits@10 obtained with the Unstructured model improves from 10.3 to 23.6, and from 32.9 to 43.0, respectively. We also always observe an improvement, although less evident, in the link prediction accuracy for the TransE model and its variants.

7. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose an unified method for leveraging schema knowledge, expressed in RDFS, in latent factor models for link prediction in knowledge graphs. We *adaptively decrease* the prediction score associated by the model to new triples, depending on whether they imply previously unknown and possibly conflicting type information. Our experimental evaluations shows that the proposed methods leads to more accurate link prediction models. Source code and datasets for reproducing the experiments are available on-line: https://github.com/knowledgegraph/schema.

Future Works. Although widely adopted, RDFS is not the only formalism for encoding schema information in KGs. We aim at devising methods for leveraging schema knowledge represented in more expressive formalisms, such as OWL 2 [9].

Acknowledgments. This work fulfills the objectives of the PON 02_00563_3489339 project "Puglia@Service - Internet-based Service Engineering enabling Smart Territory structural development" funded by the Italian Ministry of University and Research (MIUR).

8. REFERENCES

[1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - A crystallization point for the web of data. *J. Web Sem.*, 7(3):154–165, 2009.

[2] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In J. T. Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008*, pages 1247–1250. ACM, 2008.

[3] A. Bordes and E. Gabrilovich. Constructing and mining web-scale knowledge graphs: KDD 2014 tutorial. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, page 1967, 2014.

[4] A. Bordes, X. Glorot, J. Weston, and Y. Bengio. A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. *Machine Learning*, 94(2):233–259, 2014.

[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In C. Burges et al., editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc., 2013.

[6] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In S. A. Macskassy et al., editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 601–610. ACM, 2014.

[7] J. C. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[8] P. Hitzler, M. Krötzsch, and S. Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2009.

[9] D. Krompass, M. Nickel, and V. Tresp. Large-scale factorization of type-constrained multi-relational data. In *International Conference on Data Science and Advanced Analytics, DSAA 2014*, pages 18–24, 2014.

[10] F. Mahdisoltani, J. Biega, and F. M. Suchanek. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Online Proceedings*, 2015.

[11] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs: From Multi-Relational Link Prediction to Automated Knowledge Graph Construction. *CoRR*, abs/1503.00759, 2015.

[12] M. Nickel, V. Tresp, and H. Kriegel. A three-way model for collective learning on multi-relational data. In L. Getoor et al., editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 809–816. Omnipress, 2011.

[13] B. Yang, W. tau Yih, X. He, J. Gao, and L. Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, May 2015.

[9] http://www.w3.org/TR/owl2-overview/