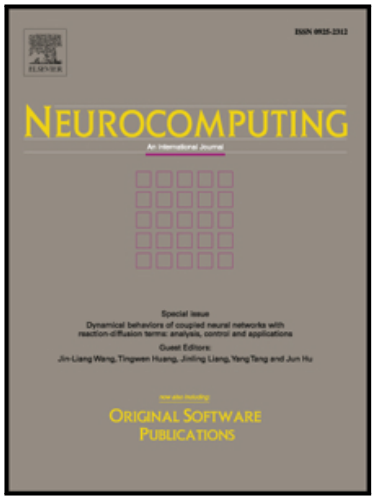
Communicated by Dr. M. Wang

Accepted Manuscript



Translating on Pairwise Entity Space for Knowledge Graph Embedding

Yu Wu, Tingting Mu, John Y. Goulermas

PII: S0925-2312(17)30796-8
DOI: 10.1016/j.neucom.2017.04.045
Reference: NEUCOM 18394

To appear in: *Neurocomputing*

Received date: 6 July 2016
Revised date: 28 November 2016
Accepted date: 18 April 2017

Please cite this article as: Yu Wu, Tingting Mu, John Y. Goulermas, Translating on Pairwise Entity Space for Knowledge Graph Embedding, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2017.04.045

Translating on Pairwise Entity Space for Knowledge Graph Embedding

Yu Wu[a], Tingting Mu[b], John Y. Goulermas[c]

[a]*Department of Electrical Engineering and Electronics, Brownlow Hill, University of Liverpool, Liverpool, L69 3GJ, UK.*

[b]*School of Computer Science, Kilburn Building, University of Manchester, Manchester, M13 9PL, UK.*

[c]*Department of Computer Science, Ashton Building, University of Liverpool, Liverpool, L69 3BX, UK.*

Abstract

In addition to feature-based representations that characterize objects with feature vectors, relation-based representations constitute another type of data representation strategies. They typically store patterns as a knowledge graph (KG), consisting of nodes (objects) and edges (relationships between objects). Given that most KGs are noisy and far from being complete, KG analysis and completion is required to establish the likely truth of new facts and correct unlikely ones based on the existing data within the KG. An effective way for tackling this, is through translation techniques which encode entities and links with hidden representations in embedding spaces. In this paper, we aim at improving the state-of-the-art translation techniques by taking into account the multiple facets of the different patterns and behaviors of each relation type. To the best of our knowledge, this is the first latent representation model which considers relational representations to be dependent on the entities they relate. The multi-modality of the relation type over different entities is effectively formulated as a projection matrix over the space spanned by the entity vectors. We develop an economic computation of the projection matrix by directly providing an analytic formulation other than relying on a more consuming iterative optimization procedure. Two large benchmark knowledge bases are used to evaluate the performance with respect to the link prediction task. A new test data partition scheme is proposed to offer better understanding of the behavior of a link prediction model. Experimental results show that the performance of the proposed algorithm is consistently among the top under different evaluation schemes.

Keywords: `Statistical relational learning`, `link prediction`, `knowledge graphs`, `hidden representation`, `embedding space.`

Email addresses: `yu.wu@liverpool.ac.uk` (Yu Wu), `tingtingmu@me.com` (Tingting Mu), `j.y.goulermas@liverpool.ac.uk` (John Y. Goulermas)

1. Introduction

There are two main ways for representing objects, of which one characterizes an object with a continuous or discrete feature vector of attributes, while the other describes the object via its relationships to other objects. The feature-based representation is usually stored as a data matrix with rows or columns corresponding to the feature vectors of different objects. Differently, the relation-based representation is usually stored as a graph, consisting of nodes (objects) and edges (relationships between objects). The edges can be labeled with different relation types or can be associated with numerical quantities. Many machine learning algorithms work directly on feature-based data representations, e.g., typical classification, clustering and ranking algorithms [1, 2, 3], or feature mapping algorithms [4, 5]. Some algorithms convert feature-based data to a graph that models pairwise neighborhood relationships among objects, and then they further process and learn from the constructed graph representation[6, 7]. Relational learning algorithms are a group of techniques specialized at handling multi-relational data [8] by processing objects interlinked by various relation types. The main resource of multi-relational data is the web-based knowledge graphs (KGs), also referred to as knowledge bases [9]. A KG stores information in a graph structured format, such as a directed graph whose nodes (entities) represent the objects and edges (links) correspond to the relation types between objects. An example of a small KG is shown in Figure 1.

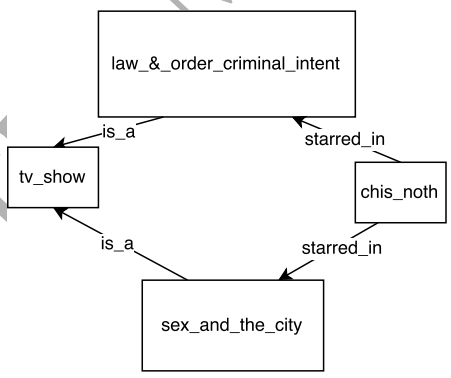


Figure 1: Real world facts stored as a KG, of which the triplet form is expressed as (head_entity, link, tail_entity), i.e. *(chris_noth, starred_in, sex_and_the_city), sex_and_the_city, is_a, tv_show).*

In the recent years, much work has been invested into the construction of large KGs, including Wordnet [10], YAGO [11], DBpedia [12], Freebase [13], NELL [14] and the Google's Knowledge Vault project[15]. These contain highly structured information and are useful in many artificial intelligence related tasks,

i.e. word-sense disambiguation [16] [17], search engine [18] [19], question answering [20]. However, despite being very large (usually containing millions of nodes and billions of edges), most KGs are very noisy and far from being complete, because large databases are either constructed collaboratively by an open group of volunteers or automatically extracted from unstructured or semi-structured text via rules, regular expressions, or machine learning and natural language processing techniques [21]. Taking Freebase as an example, which is a large collaborative knowledge base harvested from resources, such as individual and user-submitted Wiki contributions, there are 71% of around 3 million people with no known place of birth [22] within the database. Consequently, one major goal of KG analysis is to develop numerical models that suggest the likely truth of new facts and correct unlikely facts based on the existing data within KGs.

Since the KGs can correspond to massive volumes of knowledge, it is often prohibitively expensive to subject them for processing to symbolic models [23] [24] [25] or inference models [26] [27] [28] [29] [30] [31] [32] [33]. Latent representation models have therefore been receiving increasing attention. These are capable of embedding entities into a continuous vector space and converting links to mathematical operations (e.g., linear, bilinear transformation, etc.) between entity vectors with reasonable computational costs [34] [35] [36] [37][38]. TransE [39] is a representative of such models, that requires minimal parameterization and achieves very good performance. It assumes that the relationships in KGs are hierarchical and uses translations to represent them, where a single low-dimensional vector is employed to represent each targeted relationship. Its intuitive, highly scalable and effective design has driven the development of a number of translation-based algorithms [40] [41] [42] [43] [44], of which main benefits include constraining the translations within the relation-specific space and incorporating extra information (i.e., relation paths over the knowledge graphs) into the translation-based energy function.

In this work, we focus on further improving translation-based relation modeling. Our key idea is that more complex link representations could be constructed to reflect more accurately the different roles of each relation type. This is fundamentally different from the assumption made in most existing works, that only distinguish the link representations among different relation types. In real-world applications, the entity can always have exactly one meaning facilitated by the KG construction stage. However, links can be more complex and they usually correlate with each other, which makes them much harder to analyze. Therefore, more careful design is required to model link representations. Here, we show an example that the same link can possess different characters when being involved with different entity pairs, by considering the typical hierarchical link of "descendantOf". For instance, if both facts of *(person_A, descendantOf, person_B)* and *(person_B, descendantOf, person_C)* are true, *(person_A, descendantOf, person_C)* must also be true according to the hierarchical property of the "descendantOf" relationship, although it takes a longer range of dependencies than the former two triplets. It is obvious that "descendantOf" has as a direct link role with *(person_A, person_B)* and an indirect role with *(person_A,*

person_C). Existing works do not explicitly consider the different roles of the same link in different entity pairs. We propose a new translation strategy to address this, which although maps the entities and links within the same unified vector space, it models the multiple facets of each link by projecting the link vector on the relevant entity pair space to create more flexible interactions. The proposed algorithm is referred to as Translating on Pairwise Entity Space (TransPES). It is trained on a ranking based objective function using stochastic gradient descent, and is compared with multiple state-of-the-art methods in the field, using two commonly used benchmark datasets on link prediction. To facilitate a deeper analysis of the link prediction behavior, we also propose a new way for partitioning the testing relational triplets that demonstrates how the algorithm behaves on different arrangements of test data.

The remainder of this paper is organized as follows. In Section 2, a review of the previous works is provided for multi-relational learning. The mathematical formulation of our model and the associated analysis are presented in Section 3. Related experiments and evaluations are conducted in Section 4. The work is concluded along with future directions in the Section 5.

2. Related work

Early works on modelling multi-relational data employ graphical models, such as Bayesian clustering frameworks [26] [27] [28] [29] [30] or Markov logic networks [31] [32] [33]. Most of these models cannot be applied to analyze large-scale relational databases due to their high cost of inference. Another line of work treats the multi-relational data as 3-dimensional adjacency tensors, and applies tensor factorisation techniques [45] [46] [34] to analyze its link structure. One representative work is RESCAL [34], which models entities as latent feature vectors and relation types as matrices containing pairwise interaction weights between entities, and optimizes efficiently the model variables via alternating least squares. It achieves state-of-the-art accuracies for many benchmark datasets, and has been applied for link prediction on entire KGs, such as YAGO and DBpedia [11] [12].

Although the size of the adjacency tensor for modelling KGs can be very large, only a small fraction among all possible relations between entities are likely to be correct. For example, there are over 450,000 thousands actors and over 250,000 movies stored in Freebase [13], but each actor stars only in a few movies [47]. To efficiently deal with the sparse relationships in KGs, structured embedding (SE) model [35] introduces a powerful ranking loss for learning entity embeddings. This stimulates the development of a group of neural network models, such as latent factor model (LFM) [36], neural tensor networks [37], and semantic matching energy (SME) models [38], which design respective score functions to fit the likely true relations utilizing different operations between the latent entity representations. These models seem to be appropriate as they attempt to model any kind of interactions through universal numerical operations. However, they are computationally expensive and are likely to suffer

from overfitting with regard to very sparse relations, and this fails to capture intrinsic properties of the relations leading to weak model interpretability.

It has been shown in [48] that the word vectors learned from free text, coincidentally represent some hierarchical relationships as translations between word vectors; e.g., vec("Germany") + vec("capital") is closest (translated) to vec("Berlin"). This motivates the first translation-based (or called distant) model TransE [39], which is light on parameterization, but outperforms all former methods in link prediction on very large KGs. The appealing performance and scalability of this simple model has inspired the development of many others [40] [41] [42] [43] [44] that build upon the translation operations. Specifically, TransH [41] and TransR [42] assume that there is a link space for each relation type and project the entity embeddings to each link space before translation. They have shown consistent and significant improvements compared to TransE on some very large KGs. A thorough survey on relational learning techniques for analyzing KGs can be found in [9].

3. Proposed Method

A knowledge graph $\mathcal{D}$ consists of a set of links between a fixed set of entities. Let $\mathcal{E} = \{e_1, \ldots, e_{N_e}\}$ denotes the entity set and $\mathcal{R} = \{r_1, \ldots, r_{N_r}\}$ the link set. Relation information indicated by $\mathcal{D}$ can be converted to relation triplets such as (e_h, r_ℓ, e_t), where $e_h, e_t \in \mathcal{E}$ are referred as the head and tail, respectively, and $r_\ell \in \mathcal{R}$ the link (or relation type). For example, (*Champa*, *formOfgoverment*, *Monarchy*) is one of such relation triplets, where the head entity "*Champa*" and the tail entity "*Monarchy*" is linked by the relation type "*formOfgoverment*". For convenience, we denote the relation triplet (e_h, r_ℓ, e_t) as (h, ℓ, t) by referring only to the indices of the entities and links. Given a set of known links within $\mathcal{D}$, the goal is to infer unknown links and correct known but mistaken links in order to complete $\mathcal{D}$. One way to solve this task is to learn an energy function $E(h, \ell, t)$ on the set of all possible triplets in $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$, so that a triplet representing a true existing link between two entities is assembled with a low energy, otherwise with a high energy.

3.1. Motivation

Given the effectiveness, efficiency and plausible interpretability of the translation based relational learning technique, we aim to model the KG information more accurately by addressing limitations of existing works. The most commonly used translation model TransE [39] employs the following energy function

$$\mathrm{E}(h, \ell, t) = \|\boldsymbol{e}_h + \boldsymbol{r}_\ell - \boldsymbol{e}_t\|, \tag{1}$$

where $\|\cdot\|$ denotes a norm of the input vector, e.g., the Euclidean norm, and $\boldsymbol{e}_h, \boldsymbol{r}_\ell, \boldsymbol{e}_t$ are the embedding vectors of head entity, relation type and the tail entity, respectively, distributed in the same representation space. A correct relation triplet (h, ℓ, t) possesses a low energy value while an incorrect one high. This means that, in the ideal case, $\boldsymbol{e}_t$ should be the nearest neighbor of the

vector $\boldsymbol{e}_h + \boldsymbol{r}_\ell$ for a true triplet (h, ℓ, t), or should be far away from $\boldsymbol{e}_t$ for an incorrect triplet. This assumption posed by Eq. (1) can be oversimplified when processing one-to-many links. These are defined as links ℓ contained in many correct triplets (h, ℓ, t_1), (h, ℓ, t_2), ..., (h, ℓ, t_n). One example, is the "isa" link extracted from the sentence "Bob Dylan was a song writer, singer, performer, book author and film actor", based on which the following list of relation triplets can be generated

head	*link*	*tail*
(*BobDylan*,	*isa*,	*SongWriter*),
(*BobDylan*,	*isa*,	*Singer*),
(*BobDylan*,	*isa*,	*Performer*),
(*BobDylan*,	*isa*,	*BookAuthor*),
(*BobDylan*,	*isa*,	*FilmActor*).

For this type of links, TransE will return equal embeddings $\boldsymbol{e}_{t_1} = \boldsymbol{e}_{t_2} = \ldots = \boldsymbol{e}_{t_n}$ in the ideal case of zero error. Such an output fails to distinguish between different tail entities. Similarly, it can also fail to distinguish different links that are valid for the same entity pair; for instance, equal embeddings will be returned for the two different links of "presidentOf" and "placeOfbirth" to represent the two triplets of *(Obama, presidentOf, USA)* and *(Obama, placeOfbirth, USA)* in the ideal zero error case.

To overcome this shortcoming, various modifications of the above energy function have been proposed. For instance, TransM [40] allows more flexibility to model the one-to-many links by introducing a link-specific weight w_ℓ, with which the modified energy function is defined as

$$\mathrm{E}(h, \ell, t) = w_\ell \|\boldsymbol{e}_h + \boldsymbol{r}_\ell - \boldsymbol{e}_t\|. \tag{2}$$

It imposes smaller weights to one-to-many links to prevent zero error cases, so that their associated many-side entity embeddings (i.e., $\boldsymbol{e}_{t_1}, \boldsymbol{e}_{t_2}, \ldots, \boldsymbol{e}_{t_n}$ for the one-to-many link l) could possess different representations. Another modification is TransH [41], which assumes that an entity should be assigned to different representations when being involved with different links. The entity embeddings $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$ are first projected to the hyperplane of the link ℓ, denoted as $\boldsymbol{e}_h^{\perp_\ell}$ and $\boldsymbol{e}_t^{\perp_\ell}$, based on which the energy function is formulated as

$$\mathrm{E}(h, \ell, t) = \|\boldsymbol{e}_h^{\perp_\ell} + \boldsymbol{r}_\ell - \boldsymbol{e}_t^{\perp_\ell}\|_2^2. \tag{3}$$

In this case, different representations are allowed to represent the many-side embeddings for the one-to-many link l even for the zero case as long as they share the same projected representation $\boldsymbol{e}_{t_1}^{\perp_\ell} = \boldsymbol{e}_{t_2}^{\perp_\ell} = \cdots = \boldsymbol{e}_{t_n}^{\perp_\ell}$. TransR [42] further expands this idea by allowing entities and links to be distributed in different spaces of different dimensions d and k, respectively. It introduces a set of $k \times d$ projection matrices $\{\mathbf{P}_l\}_{l=1}^{N_r}$ to align the two spaces over each link, leading to the following energy function

$$\mathrm{E}(h, \ell, t) = \|\mathbf{P}_l \boldsymbol{e}_h + \boldsymbol{r}_\ell - \mathbf{P}_l \boldsymbol{e}_t\|_2^2. \tag{4}$$

In distance calculation, both TransH and TransR employ a fixed embedding representation for each link, but parameterize an entity in different ways to reflect the role difference between links, that is, $\boldsymbol{e}^{\perp_\ell}$ by TransH and $\mathbf{P}_l\boldsymbol{e}$ by TransR. However, it is more reasonable to fix the embedding representation for entities, but allow the opportunity to propagate relation information through entities. This is because of the true nature of a KG, where entity has exactly one meaning or refers to exactly one thing, but links can correlate with each other. Assume there exist entities $c_1, c_2, \ldots, d_1, d_2, \ldots, e_1, e_2, \ldots$ belonging to three classes of C, D, E, and assume that the class structure can be reflected by the link information. Another advantage of characterizing entities with fixed embedding representation is to show naturally the within-class closeness and between-class dispersion in the same space, so that it is possible to transfer the instance-based inference to the class-based inference, e.g., from $(c_i, r_1, d_j) \wedge (d_j, r_2, , e_k) \Rightarrow (c_i, r_3, e_k)$ to $(C, r_1, D) \wedge (D, r_2, E) \Rightarrow (C, r_3, E)$. A third advantage of representing entities with fixed embeddings but varying the link representation for different entity pairs, is that it offers the potential of addressing better the hierarchical relation structure. For example, the relation type like "descendentOf" can appear in multiple relation triplets such as *(person_A, descendentOf, person_B)* and *(person_B, descendentOf, person_C)*, based on which *(person_A, descendentOf, person_C)* can be inferred. Existing translation-based algorithms, as mentioned above, may not perform well to infer such relation, because their model expressive power can be limited by fixing the link representation of "descendentOf" regardless of which entity pairs it is involved with. Instead, by using different representations for "descendentOf", the model can become more flexible and formulate more accurately the interaction between"descendentOf" and different entity pairs of *(person_A, person_B)*, *(person_B, person_C)* and *(person_A, person_C)*.

3.2. The Proposed Method

3.2.1. Model Construction

The energy function of an input relation triplet is parameterized over not only three individual k-dimensional embedding vectors of its head, tail and link, but also a set of $k \times k$ transformation matrices $\{\mathbf{P}_{ht}\}_{h,t}$. Different matrices are constructed for different head-tail entity pairs (h, t) to create a bespoke link representation for a given entity pair. We formulate the energy function as

$$\mathrm{E}(h, \ell, t) = \|\boldsymbol{e}_h + \mathbf{P}_{ht}\boldsymbol{r}_\ell - \boldsymbol{e}_t\|_2, \tag{5}$$

where, apart from the l_2-norm, other ones or dissimilarity measures can be used.

To reduce the computational cost, instead of optimizing the transformation matrices, each $\mathbf{P}_{ht}$ is computed as a matrix that projects a k-dimensional vector onto the space spanned by the two (typically independent) k-dimensional entity vectors $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$. Letting the columns of the $k \times 2$ matrix $\mathbf{E}_{ht}$ be the two entity embedding vectors $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$, $\mathbf{P}_{ht}$ is then defined as the orthogonal projector

$$\mathbf{P}_{ht} = \mathbf{E}_{ht}\left(\mathbf{E}_{ht}^T\mathbf{E}_{ht}\right)^{-1}\mathbf{E}_{ht}^T. \tag{6}$$

To regularize and make the process more numerically flexible, Eq.(6) is modified according to

$$\mathbf{P}_{ht} = \mathbf{E}_{ht} \left(\mathbf{E}_{ht}^T \mathbf{E}_{ht} + \xi \mathbf{I} \right)^{-1} \mathbf{E}_{ht}^T, \tag{7}$$

where $\xi > 0$. Using Eq.(7), for sufficiently small ξ, the transformed vector $\mathbf{P}_{ht}\boldsymbol{r}_\ell$ lies very close to the entity subspace spanned by $\boldsymbol{e}_h$ and $\boldsymbol{e}_t$. This can be seen because

$$\begin{aligned}
&\mathbf{E}_{ht}^T \left(\mathbf{I} - \mathbf{P}_{ht} \right) \boldsymbol{r}_\ell \\
=&\mathbf{E}_{ht}^T \left(\boldsymbol{r}_\ell - \mathbf{E}_{ht} \left(\mathbf{E}_{ht}^T \mathbf{E}_{ht} + \xi \mathbf{I} \right)^{-1} \mathbf{E}_{ht}^T \boldsymbol{r}_\ell \right) \\
=&\mathbf{E}_{ht}^T \boldsymbol{r}_\ell - \left(\mathbf{E}_{ht}^T \mathbf{E}_{ht} + \xi \mathbf{I} - \xi \mathbf{I} \right) \left(\mathbf{E}_{ht}^T \mathbf{E}_{ht} + \xi \mathbf{I} \right)^{-1} \mathbf{E}_{ht}^T \boldsymbol{r}_\ell \\
=&\xi \left(\mathbf{E}_{ht}^T \mathbf{E}_{ht} + \xi \mathbf{I} \right)^{-1} \mathbf{E}_{ht}^T \boldsymbol{r}_\ell,
\end{aligned} \tag{8}$$

which shows that for any $\boldsymbol{r}_\ell$ we have $\lim_{\xi \to 0} \mathbf{E}_{ht}^T \left(\mathbf{I} - \mathbf{P}_{ht} \right) \boldsymbol{r}_\ell = \mathbf{0}$.

In TransR, different dimensionalities for the two embedding spaces of (d) entities and (k) links are allowed, and a set of $k \times d$ transformation matrices are employed to align the two spaces over links. Differently here, we assume equal dimensionality (k) of the two spaces, and employ a set of $k \times k$ transformation matrices to align the two spaces over entity pairs. The benefit of using equal dimensionality, is that it enables to derive an analytic form of the projection matrix as in Eq.(7) without additional effort to optimize it. For TransR, when $d > k$, the information stored in an entity embedding is compressed to a lower-dimensional vector. When $d < k$, the entity embedding is expanded to a higher-dimensional vector. However, all the expanded entities are distributed within a subspace of the link space, of which the rank of the expanded entity matrix is no more than d. Also, given the fact that the number of existing links (relation types) is usually much less than the number of existing entities in a KG, it is not necessary to increase the freedom of the link space, e.g., a higher dimensionality than the entity space. Thus, setting $d \geq k$ is more reasonable than $d < k$, and $d = k$ allows the minimal information loss, which is also the adapted setting reported in the TransR work. Because of these, we enforce equal dimensionality between the two spaces, aiming at obtaining a more mathematically convenient solution for the projection matrices without sacrificing the expressive power of the model.

3.2.2. Model Training

Given a set of known links between entities, a set of valid triplets can be constructed, which is referred to as the positive triplet set and denoted by $\mathcal{D}^+$. For each positive triplet $(h, \ell, t) \in \mathcal{D}^+$, a set of corrupted triplets can be

generated by replacing either its head or tail entity with a different one, as

$$\mathcal{D}^{-}_{h,l,t} = \left\{(h^{'},\ell,t)|h^{'} \in \{1,2,\ldots,N_e\}, (h^{'},\ell,t) \notin \mathcal{D}^{+}\right\} \cup \left\{(h,\ell,t^{'})|t^{'} \in \{1,2,\ldots,N_e\}, (h,\ell,t^{'}) \notin \mathcal{D}^{+}\right\}$$

Minimizing the energy function in Eq.(5) parameterized via the entity and link embeddings, is equivalent to the optimization of these embedding vectors to encourage the maximum discrimination between the positive and negative triplets. To achieve this, a margin-based ranking loss is employed, given as

$$\mathcal{L}_m = \sum_{(h,\ell,t)\in\mathcal{D}^{+}} \sum_{(h^{'},\ell,t^{'})\in\mathcal{D}^{-}_{h,\ell,t}} \left[\gamma + \mathrm{E}(h,\ell,t) - \mathrm{E}(h^{'},\ell,t^{'})\right]_{+}, \tag{9}$$

where $[x]_{+} \triangleq \max(0,x)$ denotes the positive part of the input x, and $\gamma > 0$ is a user-set margin parameter.

A length constraint $\|\boldsymbol{e}_i\|_2 \leq 1$ for each entity embedding is considered to prevent the training process from trivially minimizing $\mathcal{L}_m$ by arbitrarily increasing the scale of the entity embedding. This constraint can be incorporated into the cost function $\mathcal{L}_m$ as $\sum_{i=1}^{N_e}\left[\|\boldsymbol{e}_i\|_2^2 - 1\right]_{+}$. We also add a regularization term for the link embedding vectors $\{\boldsymbol{r}_j\}_{j=1}^{N_r}$. This leads to the regularized cost function

$$\mathcal{L} = \mathcal{L}_m + \lambda_1 \sum_{i=1}^{N_e}\left[\|\boldsymbol{e}_i\|_2^2 - 1\right]_{+} + \lambda_2 \sum_{j=1}^{N_r}\|\boldsymbol{r}_j\|_2^2, \tag{10}$$

where $\lambda_1 > 0$ is the scale control parameter and $\lambda_2 > 0$ is the regularization parameter. Finally, the following optimization problem is to be solved

$$\underset{\{\boldsymbol{e}_i\}_{i=1}^{N_e},\{\boldsymbol{r}_j\}_{j=1}^{N_r}}{\operatorname{argmin}} \mathcal{L}\left(\{\boldsymbol{e}_i\}_{i=1}^{N_e}, \{\boldsymbol{r}_j\}_{j=1}^{N_r}, \theta\right), \tag{11}$$

where $\theta = \{\gamma, \xi, \lambda_1, \lambda_2, k\}$ comprises the user parameter set, that includes one margin parameter, three regularization ones, and the embedding dimensionality.

The pseudocode for the proposed algorithm is provided in Algorithm 1. Similar to the optimization procedure used in [39], a stochastic gradient descent approach in minibatch mode is used. All embedding vectors for entities and relations are first initialized following the random procedure in [49]. At each main iteration, a set of positive triplets for minibatch training is randomly sampled from the training set and the corresponding corrupted triplets are generated from each individual positive triplet in this set. After a minibatch, the gradient is computed and the model parameters are updated. The algorithm terminates after a fixed number of iterations.

3.2.3. Discussion

Here we conduct some further analysis and discussion of the proposed algorithm with regard to its connections to TransE. It can be seen from Eqs.(1) and

Algorithm 1 Pseudocode for TransPES

Input: Training set $\mathcal{D} = \{(h, \ell, t)\}$, entity and link sets $\mathcal{E}$ and $\mathcal{R}$, user-provided parameter set $\theta = \{\gamma, \xi, \lambda_1, \lambda_2, k\}$, triplet minibatch of size b.

1. **Initialisation:**

   $\boldsymbol{r} \leftarrow$ uniform $(-\frac{6}{\sqrt{k}}, \frac{6}{k})$ for each $\boldsymbol{r} \in \mathcal{R}$

   $\boldsymbol{r} \leftarrow \boldsymbol{r}/\|\boldsymbol{r}\|$ for each $\boldsymbol{r} \in \mathcal{R}$

   $\boldsymbol{e} \leftarrow$ uniform $(-\frac{6}{\sqrt{k}}, \frac{6}{k})$ for each $\boldsymbol{e} \in \mathcal{E}$

   $\boldsymbol{e} \leftarrow \boldsymbol{e}/\|\boldsymbol{e}\|$ for each $\boldsymbol{e} \in \mathcal{E}$

2. **Loop:**

   $\mathcal{D}_{batch} \leftarrow$ sample from $\mathcal{D}$

   $\mathcal{T}_{batch} \leftarrow \emptyset$

   for $(h, l, t) \in \mathcal{D}$ **do**

      $(h^{'}, \ell, t^{'}) \leftarrow$ sample from $\mathcal{D}^{-}_{(h,l,t)}$

      $\mathcal{T}_{batch} \leftarrow \mathcal{T}_{batch} \cup \{((h, \ell, t), (h^{'}, \ell, t^{'}))\}$

   end for

   $\mathcal{E}_{batch} \leftarrow$ head and tail set from $\mathcal{T}_{batch}$

   $\mathcal{R}_{batch} \leftarrow$ link set from $\mathcal{T}_{batch}$

   Gradient descent update of embeddings using $\mathcal{E}_{batch}$ and $\mathcal{R}_{batch}$

   end loop

(5) that TransE formulates a true relation triplet as $\boldsymbol{r}_\ell = \boldsymbol{e}_t - \boldsymbol{e}_h$, while the proposed algorithm as $\mathbf{P}_{ht}\boldsymbol{r}_\ell = \boldsymbol{e}_t - \boldsymbol{e}_h$ to enable the modeling of more complexed relations. For instance, given three true triplets (h, ℓ, m), (m, ℓ, t) and (h, ℓ, t), a potential solution of TransE with low energy can be self-contradictory, e.g., $\boldsymbol{r}_\ell = \boldsymbol{e}_m - \boldsymbol{e}_h = \boldsymbol{e}_t - \boldsymbol{e}_m = \boldsymbol{e}_t - \boldsymbol{e}_h$ in the ideal case of zero error. By allowing different representation $\mathbf{P}_{ht}\boldsymbol{r}_\ell$ for the same link r for different entity pairs (h, t), TransPES can overcome this effect.

On the other hand, assume the relation ℓ adheres to some deterministic rules, e.g., (h, ℓ, t) can be inferred from (h, ℓ, m) and (m, ℓ, t). This transitivity pattern can be potentially modeled by using three planes H_1, H_2 and H_3, on which the projected embeddings $\boldsymbol{r}_{\ell_1}, \boldsymbol{r}_{\ell_2}, \boldsymbol{r}_{\ell_3}$ for link ℓ satisfy $\boldsymbol{r}_{\ell_1} + \boldsymbol{r}_{\ell_2} = \boldsymbol{r}_{\ell_3}$. This can be achieved by the proposed algorithm with the entities $\boldsymbol{e}_h, \boldsymbol{e}_m, \boldsymbol{e}_t$ pairwisely spanning these three planes, that is, a spanned space H_{hm} of $\boldsymbol{e}_h$ and $\boldsymbol{e}_m$ "close" to the plane H_1, H_{mt} "close" to H_2, and H_{ht} "close" to H_3. By "close", we mean that the angle between the two planes is small. Subsequently, the learned lower energies of triplets (h, ℓ, m) and (m, ℓ, t), will lead to the lower energy of (h, ℓ, t), because

$$\begin{aligned}
\|\boldsymbol{e}_h + \mathbf{P}_{ht}\boldsymbol{r}_\ell - \boldsymbol{e}_t\| &\approx \|\boldsymbol{e}_h + \boldsymbol{r}_{\ell_3} - \boldsymbol{e}_t\| \\
&= \|\boldsymbol{e}_h + \boldsymbol{r}_{\ell_1} - \boldsymbol{e}_m + \boldsymbol{e}_m + \boldsymbol{r}_{\ell_2} - \boldsymbol{e}_t\| \\
&\leq \|\boldsymbol{e}_h + \boldsymbol{r}_{\ell_1} - \boldsymbol{e}_m\| + \|\boldsymbol{e}_m + \boldsymbol{r}_{\ell_2} - \boldsymbol{e}_t\| \\
&\approx \|\boldsymbol{e}_h + \mathbf{P}_{hm}\boldsymbol{r}_\ell - \boldsymbol{e}_m\| + \|\boldsymbol{e}_m + \mathbf{P}_{mt}\boldsymbol{r}_\ell - \boldsymbol{e}_t\|. \qquad (12)
\end{aligned}$$

This indicates the possibility of encoding $(h, \ell, m) + (m, \ell, t) \Rightarrow (h, \ell, t)$ into the three spanned spaces that satisfy $\mathbf{P}_{hm}\boldsymbol{r}_\ell \approx \mathbf{P}_{hm}\boldsymbol{r}_\ell + \mathbf{P}_{mt}\boldsymbol{r}_\ell$.

Let $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n$ denote the entities that appear together with link ℓ in the true relation triplets. If one only considers to reduce the energy of correct triplets in TransE, the optimal link vector $\boldsymbol{r}_\ell^*$ must be contained in the subspace spanned by the corresponding entity embeddings. Any components added to the link embedding that are not in this subspace will increase the energy of correct triplets in TransE. However, during the training based on ranking loss, the energy of incorrect triplets is also to be maximized by seeking appropriate solution for $\boldsymbol{r}_\ell$. This will inevitably drag the learned link embedding vector $\boldsymbol{r}_\ell$ away from the optimal one $\boldsymbol{r}_\ell^*$. Differently, the proposed algorithm has the potential to learn from an incorrect triplet in the complementary space of its corresponding correct one, so that its influence over the optimal link vector $\boldsymbol{r}_\ell^*$ is automatically ignored. This enables the reduction of the energy for correct triplets and the increase of the energy for incorrect ones, simultaneously. To encourage consideration of incorrect triplets, a smaller λ_2 can be used to suppress the regularization term of $\lambda_2 \sum_{j=1}^{N_r} \|\boldsymbol{r}_j\|_2^2$ as in Eq.(10) by amplifying the effect of link embeddings.

3.3. Data Partition Scheme for Evaluation

When evaluating a link prediction task given a KG, in addition to computing an overall performance using all the test relation triplets, reseacrhers are looking

Table 1: Examples of reverse triplets.

		head	relation type	tail
E1	original	/m/012hw	/people/cause_of_death/parent_cause_of_death	/m/051_y
	reverse	/m/051_y	/base/fight/crime_type/includes_crimes	/m/012hw
E2	original	/m/0hkb8	/architecture/structure/architectural_style	/m/0f447
	reverse	/m/0f447	/architecture/architectural_style/examples	/m/0hkb8
E3	original	/m/0czp_	/award/award_category/category_of	/m/0g_w
	reverse	/m/0g_w	/award/award/category	/m/0czp_

in more detail into the different types of relation triplets and analyze how a model behaves over these different triplet types. The work in [39] suggests to group the relation triplets into the four categories of: 1-to-1, 1-to-many, many-to-1 and many-to-many, according to the cardinalities of their head and tail entities. For instance, a given triplet is classified into 1-to-many if its head entity can appear together with many tail entities in other triplets, but its tail entity only appears in this given triplet.

We propose here an alternative split of the relation triplets based on human inference logic. Specifically, it is natural for human intelligence to infer the existence of a reverse form of a given relation triplet. This can be denoted as to infer (t, ℓ^{-1}, h) from (h, ℓ, t), where ℓ^{-1} denotes the inverse link of ℓ. We list three relation triplet examples in Table 1 that appear in Freebase [13]. In each example, an original relation triplet and its reverse version that truly exist in the database are displayed, e.g., "/base/fight/crime type/includes crimes" is reverse of "/people/cause of death/parent cause of death". Another type of relation triplet that is natural for human to infer is the reciprocal relation, for which swapping the positions of the head and tail entities does not affect the validity of the relation triplet, e.g., links such as "MarriedTo" and "AliasTo". This can be denoted as to infer (t, ℓ, h) from a known triplet (h, ℓ, t) when the link ℓ is reciprocal. Taking out these two types of straightforward inference, the other inference requires more complex logic.

Our assumption is that, since human can easily infer the reverse and reciprocal triplet from the given original one, the link prediction model should be able to achieve the same. Thus, it is interesting to group the relation triplets to three categories of "reciprocal type", "reverse type", and "the other" that requires more complex logic to infer. We define the collection of known relation triplets for the model to learn from as the training set, and the testing triplets for performance evaluation as the test set. The following split is applied to the test set: If a testing triplet (h, ℓ, t) is reciprocal, (t, ℓ, h) should be found in the training set. If a test triplet (h, ℓ, t) belongs to the reverse type, its reverse form (t, ℓ^{-1}, h) should appear in the training set. However, it is not easy to identify the reverse relation for any given relation type due to the lack of information. So we relax the condition to that, if (h, ℓ, t) is a reverse type, $(t, *, h)$ should exist in the training set without specifying the involved link. After identifying the reciprocal and reverse triplets, the remaining ones in the test set are categorized as "the others". Individual evaluation over each category of the testing triplet provides deeper insight on the studied model.

4. Experiments

4.1. Datasets and Experimental Setup

The proposed algorithm is compared with ten state-of-the-art translation models from the literature (see Table 3), evaluated using two benchmark link prediction datasets of WN18 [38] and FB15k [35] extracted from the two large real-world knowledge bases of Wordnet [10] and Freebase [13], respectively. We provide below some brief description of WN18 and FB15K datasets, and show their statistics in Table 2.

- The *WN18* dataset contains a total of 40,943 entities, 18 relational types and 151,442 relation triples. It is extracted from the large English lexical database Wordnet, which groups words into sets of cognitive synonyms (synsets) and interlinks these synsets by means of a small number of semantic relations, such as *synonymy*, *antonymy*, *meronymy* and *hypernymy*. One example of a typical triplet is *(_range_NN_3, _hypernym, _tract_NN_1)*, which means the third meaning of the noun "range" is a hypernym of the first sense of the noun "tract".

- The *FB15k* dataset contains a total of 14,951 entities, 1345 relation types and 592,213 relation triples. It is created by adopting the frequently occurrent entities and relationships in Freebase, which is a massive online collection database consisting of general human knowledge. It organises the human knowledge data directly in the triplet form of *(head, link, tail)*. Typical triplet examples are *(Peasant Magic, album, Solipsitalism)*, *(Barack Obama, religion, Christianity)* and *(Orange France, place-founded, Paris)*.

Table 2: Statistics of datasets

Dataset	WN18	FB15k
Relationships	18	1,345
Entities	40,943	14,951
Train	141,442	483,142
Valid	5,000	50,000
Test	5,000	59,071

The proposed algorithm is compared with ten state-of-the-art translation models in terms of link prediction performance. Essentially, every model is trained by optimizing a score function (or an energy function in our case) to assemble the likely relation triples with higher scores (or lower energies) than the unlikely relations. This function can thus give its estimation of the likely score (or energy) for every true triplet in the test set. The following evaluation metrics based on the predicted score (or energy) are used:

- *Mean rank* measures how well the predicted scores (or energies) correlate with the true triplets [35]. For each correct triplet in the test set, we first construct the corrupt triples by replacing the head entity with all the

entities in the knowledge base. The scores (or energies) of these corrupted triples are first computed and then sorted in descending (or ascending for energy) order, and the rank for the correct head entity is stored. This procedure is repeated by replacing the tail entity of each correct triple with all the entities in the knowledge base to obtain the rank for each correct tail entity. The average of all these predicted ranks in the test set is used to report the performance.

- *Hits@10* is another measure of the correlation between the predicted scores (or energies) and the true triplets [35]. Following exactly the same ranking procedure as in the mean rank evaluation, hits@10 is the proportion of the correct triplets ranked within top 10 of all the evaluated ones.

Previous work [39] suggests to filter out corrupted triplets that appear to be valid ones in the given triplets (for all the training, validation and test sets), as they should not be counted as errors. We conduct this filtering procedure to calculate filtered mean rank and hits@10 performance. To distinguish performance computed with and without the filtering procedure, we refer it as *raw* without filtering and *filtered* with the filtering procedure.

The same training, validation and test splits provided by [39] are used to evaluate the proposed model. The resulting performance is compared against performance of the state-of-the-art models that is reported in the literature [39] [41] [42] using their recommended settings as stated in the papers. Parameters of TransPES were tuned using validation set based on simple grid searches. The learning rate was searched among $\{0.1, 0.01, 0.002\}$, the dimension of the entity and link embedding k among $\{20, 50, 100\}$, the regularization parameter for scaling control λ_1 was assigned as a constant value 1, batch size B among $\{50, 100, 200\}$, the regularization parameter λ_2 among {0.1,0.01,0.001} and the margin γ betwen 0 and 1 with step size of 0.1. The regularization parameter ξ is fixed as a small positive value 10^{-8}. For both datasets, the epochs round was set as not more than 1000 times. The best model among the last 100 epochs was selected according to the mean rank and hits@10 performance of the validation set. An open-source implementation of TransPES is available from the project webpage[1]. The optimal configurations returned by the searching procedure are $k = 20, B = 100, \lambda_2 = 0.01, \gamma = 0.7$ for WN18 and $k = 100, B = 100, \lambda_2 = 0.01, \gamma = 0.4$ for FB15k.

4.2. Performance Comparison

Performance of the proposed and competing methods are reported in Table 3. The proposed TransPES provides in general better performance than the competing ones, particularly for the larger and more complex dataset FB15k containing $1,345$ relation types. Although TransR provides good performance for the smaller dataset WN18, it performs less wells for the larger dataset FB15K.

[1] `https://github.com/while519/TranPES.git`

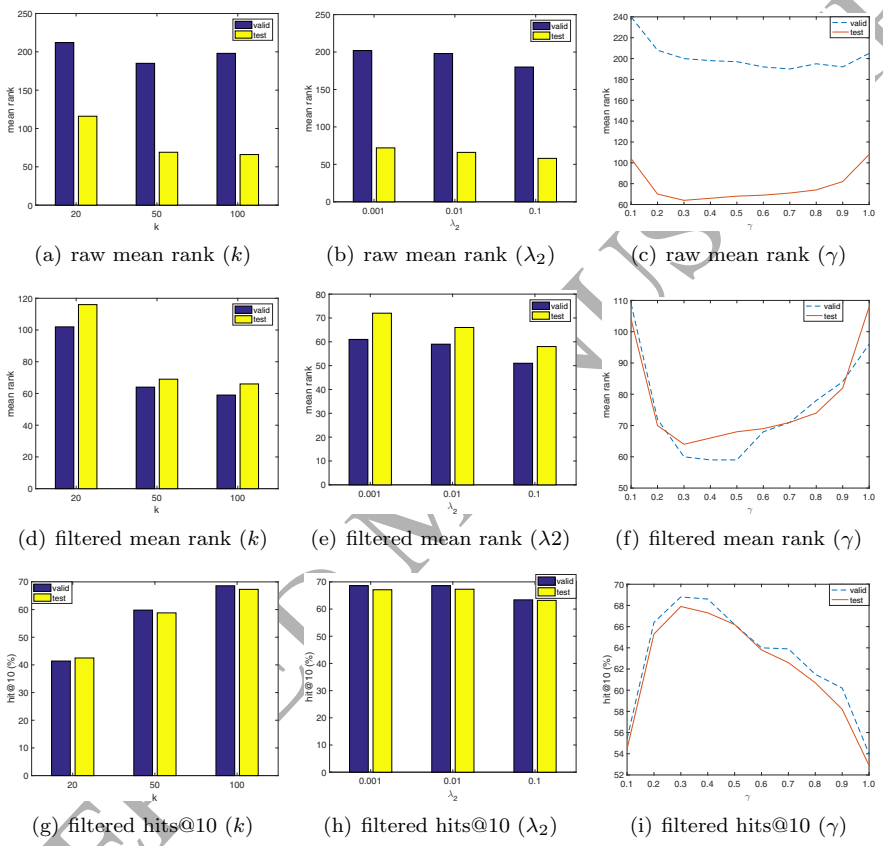
In terms of optimization complexity, TransR requires to opitmize much more variables than TransE and TransPES. We also demonstrate how the TransPES performance changes against different settings of the embedding dimensionality (k), regularization parameter (λ_2) and margin parameter (γ) using the FB15K dataset. In each implementation, two parameters are fixed as the ones in the optimal configuration, different settings of the third parameter within the searching range are examined, for which the raw and filtered mean ranks, also the filtered hits@10 performance for both the validation and test sets are reported in Figure 2. It can be seen that TransPES is less sensitive to the regularization parameter λ_2 than to the embedding dimension k and margin parameter γ.

We further analyze the performance of the large dataset FB15K in detail using the detailed evaluation protocol suggested in [39], which classifies the hits@10 results according to four categories of relationship including 1-to-1 (1-1), 1-to-many (1-M), many-to-1 (M-1) and many-to-many (M-M).The corresponding results are shown in Table 4. It can be seen from the table that the proposed algorithm consistently outperforms most the competing ones, provides similarly good performance as the cluster-based TransR (CTransR). As expected, TransPES provides satisfactory performance to predict head entity in the 1-to-1, 1-to-many relationships and predict tail entity in the 1-to-1 and many-to-1 relationships.

We conduct deeper analysis for the FB15k dataset using the proposed evaluation scheme as explained in Section 3.3, based on which 4,336 (7.3%) reciprocal triplets and 41,481 (70.2%) reverse triplets are identified, and the remaining triplets correspond to "the others" type. Most test triplets can find their reciprocal or repetitive forms in the training set to support the inference. In Table 5, we compare the TransE and TransPES performance by examining how well they infer the reciprocal and reverse type of triples in the test set in Table 5. It can be seen from the table that the proposed algorithm achieves much better results (58.8% vs. 82.1% on the reciprocal triplets and 56.9% vs. 72.4% on the reverse ones). On the other hand, for the more challenging triplets of "the other" type, both algorithms experience a very large decrease in the performance.

Table 3: Performance comparison for WN18 and FB15k datasets. The best performance is highlighted in bold, and second best underlined.

Dataset	WN18				FB15k			
Metric	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filtered	Raw	Filtered	Raw	Filtered
Unstructured [38]	315	304	35.3	38.2	1,074	979	4.5	6.3
RESCAL [34]	1,180	1,163	37.2	52.8	828	683	28.4	44.1
SE [35]	1,011	985	68.5	80.5	273	162	28.8	39.8
SME(linear) [38]	545	533	65.1	74.1	274	154	30.7	40.8
SME(bilinear) [38]	526	509	54.7	61.3	284	158	31.3	41.3
LFM [36]	469	456	71.4	81.6	283	164	26.0	33.1
TransE [39]	263	251	75.4	89.2	243	125	34.9	47.1
TransH [41]	318	303	75.4	86.7	<u>211</u>	84	42.5	58.5
TransR [42]	<u>232</u>	<u>219</u>	<u>78.3</u>	<u>91.7</u>	226	<u>78</u>	43.8	65.5
CTransR [42]	243	230	**78.9**	**92.3**	233	82	<u>44</u>	<u>66.3</u>
TransPES	**223**	**212**	71.6	81.3	**198**	**66**	**48.05**	**67.3**



(a) raw mean rank (k) (b) raw mean rank (λ_2) (c) raw mean rank (γ)

(d) filtered mean rank (k) (e) filtered mean rank ($\lambda 2$) (f) filtered mean rank (γ)

(g) filtered hits@10 (k) (h) filtered hits@10 (λ_2) (i) filtered hits@10 (γ)

Figure 2: Illustration of the performance change of TransPES against each of its three algorithm parameters (k, λ_2, γ) in terms of the raw and filtered mean rank, also the filtered hits@10 measurements, evaluated using validation and test sets marked as "valid" and "test" respectively in each plot.

Table 4: Detailed Evaluation on FB15k. Best performance is highlighted in bold.

Tasks	Predicting Head (Hits@10)				Predicting Tail (Hits@10)			
Relation Category	1-1	1-M	M-1	M-M	1-1	1-M	M-1	M-M
Unstructured [38]	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE [35]	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME(linear) [38]	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME(bilinear) [38]	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE [39]	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH [41]	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransR [42]	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
CTransR [42]	**78.6**	77.8	36.4	**68.0**	77.4	37.8	78.0	**70.3**
TranPES	78.0	**88.6**	**38.9**	67.3	**78.9**	**42.1**	**84.2**	69.8

Table 5: Link prediction comparison between TransE and TransPES over the reciprocal, reverse and other triplets in the test set of FB15k data.

Methods	TransE		TranPES	
Metrics	MAR	Hits@10(%)	MAR	Hits@10 (%)
Reciprocal	46	58.8	10	82.1
Reverse	75	56.9	28	72.4
Others	157	48.9	204	46.6

5. Conclusion

We have presented a new translation-based relational learning algorithm to encode relation triplets in KGs using link and entity embeddings, under the constraint of employing simple operations, such as vector addition and projection to encode interlinkages in KGs and maintain very low computational cost and better model interpretability. Facing the challenge of accurately modeling complex relation logic via simple operations, the key is to unfold the relation logic by determining appropriate subspaces to work on. The proposed TransPES allows multiple representations for a single relation type to model its multimodality behavior when interacting with different entity pairs, and employs fixed embedding representation for entities to permit smooth propagation of information across the graph. Interactions between links and entities are formulated in different spaces spanned by different entity pairs to offer bespoke link presentation for a targeted entity pair. Performance comparison with state-of-the-art methods and deep analysis of the algorithm behavior based on different test data partitions demonstrate the superiority of the proposed algorithm.

Acknowledgment

This research was supported by a PhD studentship, jointly funded from the University of Liverpool and the China Scholarships Council.

References

[1] S. R. Kulkarni, G. Lugosi, and S. S. Venkatesh, "Learning pattern classification-a survey," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2178–2206, 1998.

[2] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–279, 2014.

[3] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for web image search," *IEEE Trans. Image Processing*, vol. 21, no. 11, pp. 4649–4661, 2012.

[4] C. Luo, B. Ni, S. Yan, and M. Wang, "Image classification by selective regularized subspace learning," *IEEE Trans. Multimedia*, vol. 18, no. 1, pp. 40–50, 2016.

[5] M. Wang, W. Li, D. Liu, B. Ni, J. Shen, and S. Yan, "Facilitating image search with a scalable and compact semantic mapping," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1561–1574, 2015.

[6] M. Wang, X. Liu, and X. Wu, "Visual classification by ℓ_1-hypergraph modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2564–2574, 2015.

[7] M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu, "Scalable semi-supervised learning by efficient anchor graph regularization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, 2016.

[8] R. Davis and H. Shrobe, "Mit ai lab and symbolics, inc. peter szolovits mit lab for computer science," *AI Magazine*, vol. 14, no. 1, pp. 17–33, 1993.

[9] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, 2016.

[10] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995. [Online]. Available: http://dl.acm.org/citation.cfm?id=219748

[11] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 697–706. [Online]. Available: http://dl.acm.org/citation.cfm?id=1242667

[12] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and others, "DBpediaa large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015. [Online]. Available: http://content.iospress.com/articles/semantic-web/sw134

[13] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 1247–1250. [Online]. Available: http://dl.acm.org/citation.cfm?id=1376746

[14] J. Betteridge, A. Carlson, S. A. Hong, E. R. Hruschka Jr, E. L. Law, T. M. Mitchell, and S. H. Wang, "Toward Never Ending Language Learning." in *AAAI Spring Symposium: Learning by Reading and Learning to Read*, 2009. [Online]. Available: http://www.aaai.org/Papers/Symposia/Spring/2009/SS-09-07/SS09-07-001.pdf

[15] 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 *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 601–610. [Online]. Available: http://dl.acm.org/citation.cfm?id=2623623

[16] V. Ng and C. Cardie, "Improving machine learning approaches to coreference resolution," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 104–111.

[17] R. S. S. Prakash, D. Jurafsky, and A. Y. Ng, "Learning to merge word senses," *EMNLP-CoNLL 2007*, vol. 1005, 2007.

[18] "Introducing the Knowledge Graph: things, not strings." [Online]. Available: https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

[19] "Understand Your World with Bing." [Online]. Available: http://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/

[20] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager *et al.*, "Building watson: An overview of the deepqa project," *AI magazine*, vol. 31, no. 3, pp. 59–79, 2010.

[21] G. Weikum and M. Theobald, "From information to knowledge: harvesting entities and relationships from web sources," in *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2010, pp. 65–76. [Online]. Available: http://dl.acm.org/citation.cfm?id=1807097

[22] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin, "Knowledge Base Completion via Search-Based Question Answering," 2014. [Online]. Available: http://research.google.com/pubs/pub42024.html

[23] J. Pearl and A. Paz, *Graphoids: A graph-based logic for reasoning about relevance relations.* University of California (Los Angeles). Computer Science Department, 1985. [Online]. Available: http://ftp.cs.ucla.edu/pub/stat_ser/r53-L.pdf

[24] R. Fagin, J. Y. Halpern, Y. Moses, and M. Vardi, *Reasoning about knowledge.* MIT press, 2004. [Online]. Available: https://books.google.co.uk/books?hl=en&lr=&id=hvDuCwAAQBAJ&oi=fnd&pg=PR7&dq=reasoning+knowledge+graph&ots=ouTzwA7d5Q&sig=eSfmKml530RDK6VEdvRa98sxHP8

[25] J. F. Sowa, "Conceptual graphs," *Foundations of Artificial Intelligence*, vol. 3, pp. 213–237, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574652607030052

[26] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *IJCAI*, vol. 99, 1999, pp. 1300–1309. [Online]. Available: http://www.robotics.stanford.edu/~koller/Papers/Friedman+al:IJCAI99.pdf

[27] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, "Learning systems of concepts with an infinite relational model," in *AAAI*, vol. 3, 2006, p. 5. [Online]. Available: http://www.aaai.org/Papers/AAAI/2006/AAAI06-061.pdf

[28] Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel, "Learning infinite hidden relational models," *Uncertainity in Artificial Intelligence (UAI2006)*, 2006. [Online]. Available: https://pdfs.semanticscholar.org/f7bc/53641e31317fdd091d5cc148f90cb4f7c9cc.pdf

[29] K. Miller, M. I. Jordan, and T. L. Griffiths, "Nonparametric latent feature models for link prediction," in *Advances in neural information processing systems*, 2009, pp. 1276–1284. [Online]. Available: http://papers.nips.cc/paper/3846-nonparametric-latent-feature-models-for-link-prediction

[30] I. Sutskever, J. B. Tenenbaum, and R. R. Salakhutdinov, "Modelling relational data using bayesian clustered tensor factorization," in *Advances in neural information processing systems*, 2009, pp. 1821–1828. [Online]. Available: http://papers.nips.cc/paper/3863-modelling-relational-data-using-bayesian-clustered-tensor-factorization

[31] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1-2, pp. 107–136, 2006. [Online]. Available: http://link.springer.com/article/10.1007/s10994-006-5833-1

[32] S. Kok and P. Domingos, "Statistical predicate invention," in *Proceedings of the 24th international conference on Machine learning.* ACM, 2007, pp. 433–440. [Online]. Available: http://dl.acm.org/citation.cfm?id=1273551

[33] P. Singla and P. Domingos, "Entity resolution with markov logic," in *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 2006, pp. 572–582. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4053083

[34] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 809–816. [Online]. Available: http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2011Nickel_438.pdf

[35] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Conference on Artificial Intelligence*, 2011. [Online]. Available: http://infoscience.epfl.ch/record/192344/files/Bordes_AAAI_2011.pdf

[36] R. Jenatton, N. L. Roux, A. Bordes, and G. R. Obozinski, "A latent factor model for highly multi-relational data," in *Advances in Neural Information Processing Systems*, 2012, pp. 3167–3175. [Online]. Available: http://papers.nips.cc/paper/4744-a-latent-factor-model-for-highly-multi-relational-data

[37] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning With Neural Tensor Networks for Knowledge Base Completion," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 926–934. [Online]. Available: http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion.pdf

[38] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233–259, Feb. 2014. [Online]. Available: http://link.springer.com/article/10.1007/s10994-013-5363-6

[39] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems*, 2013, pp. 2787–2795. [Online]. Available: http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data

[40] M. Fan, Q. Zhou, E. Chang, and T. F. Zheng, "Transition-based knowledge graph embedding with relational mapping properties," in *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, 2014, pp. 328–337. [Online]. Available: http://anthology.aclweb.org/Y/Y14/Y14-1039.pdf

[41] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the Twenty-Eighth AAAI*

Conference on Artificial Intelligence. Citeseer, 2014, pp. 1112–1119. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.486.2800&rep=rep1&type=pdf

[42] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of AAAI*, 2015. [Online]. Available: http://166.111.138.24/~lzy/publications/aaai2015_transr.pdf

[43] A. Garca-Durn, A. Bordes, and N. Usunier, "Composing Relationships with Translations," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 286–290. [Online]. Available: http://www.aclweb.org/anthology/D15-1034.pdf

[44] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," *arXiv preprint arXiv:1506.00379*, 2015. [Online]. Available: http://arxiv.org/abs/1506.00379

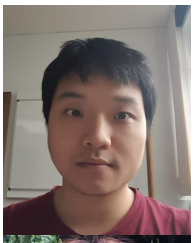
[45] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 650–658. [Online]. Available: http://dl.acm.org/citation.cfm?id=1401969

[46] T. Franz, A. Schultz, S. Sizov, and S. Staab, "Triplerank: Ranking semantic web data by tensor decomposition," in *International semantic web conference*. Springer, 2009, pp. 213–228. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-04930-9_14

[47] 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," *arXiv preprint arXiv:1503.00759*, 2015. [Online]. Available: http://arxiv.org/abs/1503.00759

[48] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119. [Online]. Available: http://papers.nips.cc/paper/5021-distributed-representations

[49] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256. [Online]. Available: http://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2010_GlorotB10.pdf



Yu Wu is currently a Ph.D. student at the University of Liverpool. He obtained his B.Eng. degree in Mechanical Engineering and Automation from the University of Science and Technology of China, Hefei, China, in 2012. His research interests include machine learning and multimedia data analysis, such as large-scale multimedia indexing and retrieval, and multimedia data embedding.

Tingting Mu received the B.Eng. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2004, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool in 2008. She is currently a Lecturer in the Department of Computer Science at the University of Manchester. Her current research interests include machine learning, data visualization and mathematical modeling, with applications to information retrieval, text mining, and bioinformatics.

John Y. Goulermas obtained the B.Sc.(1st class) degree in computation from the University of Manchester (UMIST), in 1994, and the M.Sc. and Ph.D. degrees from the Control Systems Center, UMIST, in 1996 and 2000, respectively. He is currently a Reader in the Department of Computer Science at the University of Liverpool. His current research interests include machine learning, combinatorial data analysis, data visualization as well as mathematical modeling. He has worked with various application areas including image/video analysis, biomedical engineering and biomechanics, industrial monitoring and control, and security. He is a senior member of the IEEE and an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems.