

Large Scale Graph Mining

MULTI-RELATIONAL GRAPH EMBEDDING APPROACHES

Nacéra Seghouani

Computer Science Department, CentraleSupélec
Laboratoire Interdisciplinaire des Sciences du Numérique, LISN
nacera.seghouani@centralesupelec.fr

2024-2025

Multi-relational and Knowledge Graphs

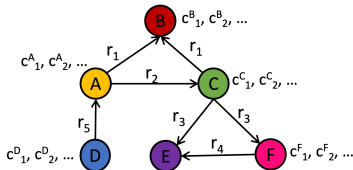
- Multi-relational graph refers to a directed graph $\mathcal{G}(E, R)$ whose nodes are entities $\in E$ and edges are relationships between entities $\in R$.
- $\mathcal{G}$ could be represented as a set of triples of head h , tail t and their relationship r

$$\{(h, r, t) | h, t \in E \text{ and } r \in R\}$$

*

Nodes: entities of different classes $c_1, c_2, \dots$

Edges: relationships between entities of different types: r_1, r_2, r_3, r_4 and r_5



Multi-relational and Knowledge Graphs



Examples of multi-relational data:

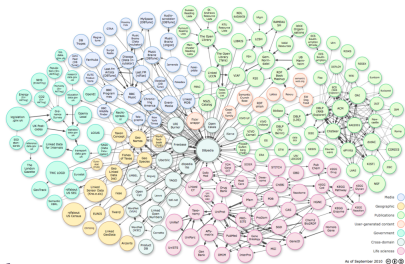
- Social network (Facebook, LinkedIn) , Recommender systems (Amazon Product Graph), ...
- Knowledge graphs: DBpedia, Google Knowledge Graph, IBM Watson, Microsoft Satori, Freebase, Wikidata, Yago, Nell or GeneOntology

Some of them are publicly available (linked open data)



Knowledge graph is a multi-relational graph where the semantics underlying the entities and their relationships (Abox) are formally described in a terminological box (Tbox)

- reasoning capabilities based on logical facts (axioms and rules)
- W3C languages for representation and querying such graphs
- Some examples: classes of entities, subclassOf between classes, domain/range of properties, subproperty between properties ...



As of September 2011

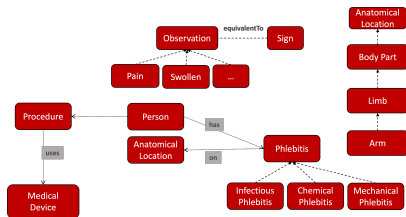
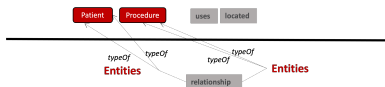


Multi-relational and Knowledge Graphs

- Knowledge graph is a multi-relational graph where the semantics underlying the entities and their relationships (Abox) are formally described in a terminological box (Tbox)
 - reasoning capabilities based on logical facts (axioms and rules)
 - W3C languages for representation and querying such graphs
 - Some examples: classes of entities, subclassOf between classes, domain/range of properties, subproperty between properties ...

A knowledge graph-structured data to represent entities, their relationships while also encoding the semantics underlying these entities.

Classes, Properties Triples + Rules



Multi-relational and Knowledge Graphs

☞ Examples of multi-relational data:

- Social networks (Facebook, LinkedIn) , Recommender systems (Amazon Product Graph), ...
- Knowledge graphs: DBPedia, Google Knowledge Graph, IBM Watson, Microsoft Satori, Freebase, Wikidata, Yago, Nell or GeneOntology

Some of them are publicly available (linked open data)

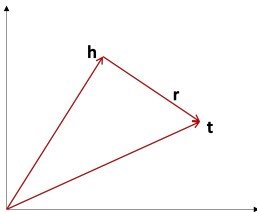
☞ Applications: Question answering/conversational agents, recommendation systems, search engines, ...

☞ Common characteristics: massive, noisy and incomplete

☞ Problem: how to embed entities and relationships of multi-relational data in low-dimensional vector spaces. How to define similar entities using triples?

Multi-relational Graph Embedding: TransE

- Translational methods (also shallow embedding) such as TransE, TransH, TransR, ... and many others DistMult, ComplEx, and RotatE, TransOWL
- TransE (Bordes et al. [↗](#)) canonical model which is easy to train, contains a reduced number of parameters and can scale up to very large databases.
- TransE represents both entities and relations as vectors in the same space, $\mathbb{R}^d$, d is a hyperparameter.
 - ✓ Given $(h \in E, r \in R, t \in E)$, the relation embedding $\mathbf{r}$ is interpreted as a translation vector from head embedding $\mathbf{h}$ to tail embedding $\mathbf{t}$ st. $\mathbf{t} \approx \mathbf{h} + \mathbf{r}$
 - ✓ For $(\text{AlfredHitchcock}, \text{DirectorOf}, \text{Psycho})$, $(\text{JamesCameron}, \text{DirectorOf}, \text{Avatar})$ the intuition is to capture regularities such that $(\text{Psycho} - \text{AlfredHitchcock} \approx \text{Avatar} - \text{JamesCameron})$ through this relation we can get $\text{AlfredHitchcock} + \text{DirectorOf} \approx \text{Psycho}$ and $\text{JamesCameron} + \text{DirectorOf} \approx \text{Avatar}$.

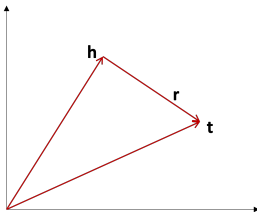


Multi-relational Graph Embedding: TransE

- TransE represents both entities and relations as vectors in the same space, $\mathbb{R}^d$, d is a hyperparameter.
- The loss function computes the similarity score between $\mathbf{h}$ translated by $\mathbf{r}$ and $\mathbf{t}$, using the norm L_2 .

$$f_r(h, t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2$$

✓ expected to be large if (h, r, t) holds.



Multi-relational Graph Embedding: TransE

- Contrastive loss: favors lower distance (or higher score) for valid triplets, high distance (or lower score) for corrupted ones
- Given a training set $\mathcal{S}^+$ of triples $(h \in E, r \in R, t \in E)$, and the set $\mathcal{S}^-$ of corrupted triples constructed according to:

$$\mathcal{S}_{(h,r,t)}^- = \{(h', r, t) | h' \in E\} \cup \{(h, r, t') | t' \in E\}$$

with either the head or tail replaced by a random entity (but not both at the same time).

- To learn such embeddings, a loss function, a margin-based ranking over the training set:

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{S}^+} \sum_{(h',r,t') \in \mathcal{S}_{(h,r,t)}^-} [\gamma + f_r(h, t) - f_r(h', t')]$$

- ✓ $\gamma > 0$ is a margin hyperparameter. The loss f favors lower values of the energy for training triplets than for corrupted triplets.
- ✓ Note that for a given entity, its embedding vector is the same when the entity appears as the head or as the tail of a triplet.
- ✓ The optimization is carried out by SGD, with constraints that the L_2 -norm of the embeddings of the entities is 1, to prevent to trivially minimize $\mathcal{L}$ by artificially increasing entity embeddings norms.

Multi-relational Graph Embedding: TransE

input: $S^+ = \{(h, r, t)\}$, E , R , dimension d , margin γ

output: Embedding vectors $\mathbf{e}, \mathbf{r}$

initialization: Embedding vectors $\mathbf{e}, \mathbf{r}$ are randomly uniformly initialized, and normalised

$\mathbf{r} \leftarrow \text{uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}}), \mathbf{r} \leftarrow \frac{\mathbf{r}}{\|\mathbf{r}\|} \mathbf{r} \in R, \mathbf{r} \in \mathbb{R}^{|R|}$

$\mathbf{e} \leftarrow \text{uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}}), \mathbf{e} \in E$

while stop condition **do**

$\mathbf{e} \leftarrow \frac{\mathbf{e}}{\|\mathbf{e}\|}, \mathbf{e} \in \mathbb{R}^{|E|}$ /*entity embedding norm is 1, to prevent to trivially minimize $\mathcal{L}$ by artificially increasing entity embeddings norms.*/*

$S_{batch} \leftarrow \text{sample}(S^+, b)$ /*From the training set randomly sample a batch of size b */

for each $(h, r, t) \in S_{batch}$ **do**

$(h', r, t') \leftarrow \text{sample}(S^-)$ /*sample a corrupted fact of triple*/

$T_{batch} \leftarrow T_{batch} \cup \{((h, r, t), (h', r, t'))\}$

end for

 Update embeddings by minimizing the loss function using GDS

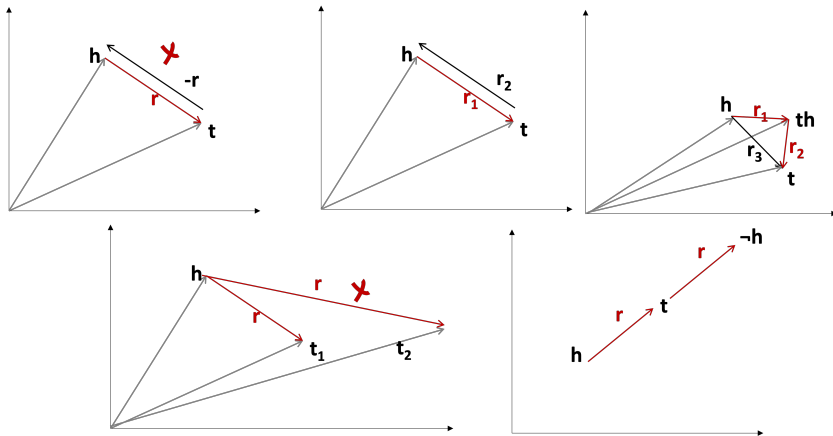
end while

Multi-relational Graph Embedding: TransE

✓ What is the behavior of TransE given the properties of the relations? Is TransE expressive enough to model these relations?

- ☞ Symmetry: $(h, r, t) \in \mathcal{G} \Rightarrow (t, r, h) \in \mathcal{G}$. Examples of r : roommate, brother, ...
- ☞ Inverse: $(h, r_1, t) \in \mathcal{G} \Rightarrow (t, r_2, h) \in \mathcal{G}$. Examples of r_1 and r_2 : advisor and advisee, parent and child, husband and wife ...
- ☞ Composition: $(h, r_1, th), (th, r_2, t) \in \mathcal{G}$ and $\Rightarrow (h, r_3, t) \in \mathcal{G}$. Cases where $r_1 = r_2$ or r_3
- ☞ Transitive: $(h, r, th), (th, r, t) \in \mathcal{G}$ and $\Rightarrow (h, r, t) \in \mathcal{G}$. Examples of r : sibling, ..
- ☞ 1-to- N : $(h, r, t_1) \in \mathcal{G} \Rightarrow \exists (h, r, t_2) \in \mathcal{G}$. Examples of r : mother, father ..
- ☞ Antisymmetry: $(h, r, t) \in \mathcal{G} \Rightarrow \neg (t, r, h) \in \mathcal{G}$. Ex. : parent, ...

Multi-relational Graph Embedding: TransE



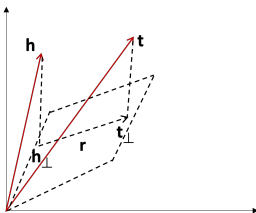
⇒ The representation of an entity is the same when involved in any relations, ignoring distributed representations.

Multi-relational Graph Embedding: TranH

- Despite its simplicity and efficiency, TransE has flaws in dealing with 1-to- N or N -to- N relations. For *DirectorOf* relation TransE might learn very similar vector representations for *Psycho*, *Rebecca* and *RearWindow* which are all related to *AlfredHitchcock* even though they are totally different entities.
- To overcome the disadvantages: an entity have different representations when involved in different relations. TransH introduced relation-specific hyperplanes .

$$f_r(h, t) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2 \text{ with } \mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^T \mathbf{h} \mathbf{w}_r \text{ and } \mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^T \mathbf{t} \mathbf{w}_r$$

where each $\mathbf{r}$ is a vector on a hyperplane with $\mathbf{w}_r$ is the projection normal vector

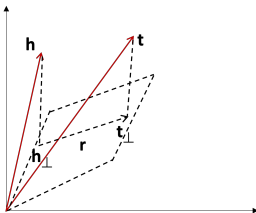


Multi-relational Graph Embedding: TranH

- Despite its simplicity and efficiency, TransE has flaws in dealing with 1-to- N or N -to- N relations. For *DirectorOf* relation TransE might learn very similar vector representations for *Psycho*, *Rebecca* and *RearWindow* which are all related to *AlfredHitchcock* even though they are totally different entities.
- To overcome the disadvantages: an entity have different representations when involved in different relations. TransH introduced relation-specific hyperplanes $\square$.

$$f_r(h, t) = -\|\mathbf{h}_{\perp} + \mathbf{r} - \mathbf{t}_{\perp}\|_2^2 \text{ with } \mathbf{h}_{\perp} = \mathbf{h} - \mathbf{w}_r^T \mathbf{h} \mathbf{w}_r \text{ and } \mathbf{t}_{\perp} = \mathbf{t} - \mathbf{w}_r^T \mathbf{t} \mathbf{w}_r$$

where each $\mathbf{r}$ is a vector on a hyperplane with $\mathbf{w}_r$ is the projection normal vector



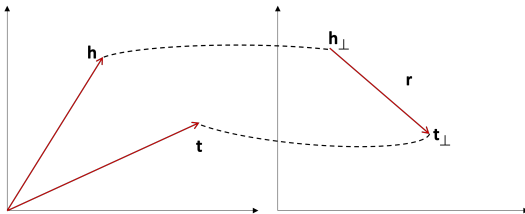
⇒ Show how TransH represent relation properties presented before?

Multi-relational Graph Embedding: TransR

- TransR shares a very similar idea with TransH but introduces a relation specific spaces $\mathbb{R}^k$. In TransR, entities are represented as vectors in an entity space $\mathbb{R}^d$, and each relation is associated with a specific space $\mathbb{R}^k$ and modeled as a translation vector in that space.
- Powerful in modelling complex relations. it introduces a projection matrix for each relation $O(dk)$ parameters for each relation whereas TransE and TransH require $O(d)$ per relation.

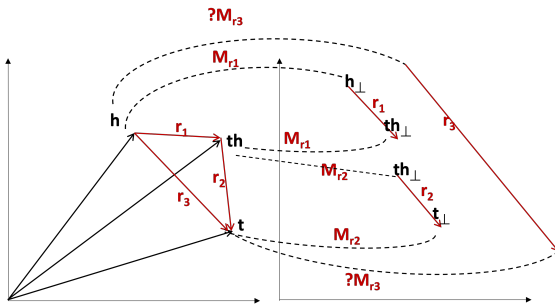
$$f_r(h, t) = -||\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp||_2^2 \text{ with } \mathbf{h}_\perp = \mathbf{M}_r \mathbf{h}, \mathbf{t}_\perp = \mathbf{M}_r \mathbf{t}$$

where each $\mathbf{M}_r \in \mathbb{R}^{k \times d}$ is a projection matrix from the entity space to the relation space of r .



Multi-relational Graph Embedding: TransR

- ☞ Show how TransR represent symmetry, inverse, transitive and one-to-many properties?
- ☞ Note that $\mathbf{M}_r \neq \mathbf{M}_{-r}$ and $\mathbf{M}_{r_1} \neq \mathbf{M}_{r_2}$
- ☞ Transitive $(h, r, th), (th, r, t) \in \mathcal{G}$ and $\Rightarrow (h, r, t) \in \mathcal{G}$: need to know if $\mathbf{M}_{r_3}$ exists given $\mathbf{M}_{r_1}$ and $\mathbf{M}_{r_2}$
- ☞ What about reflexive relation in TransE, TransH and TransR?



Multi-relational Graph Embedding

Method	Entity	Relation	scoring function	Regularisation	# parameters
TransE	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\ \mathbf{h} + \mathbf{r} - \mathbf{t}\ _2$	$\ \mathbf{h}\ _2 = 1, \ \mathbf{t}\ _2 = 1$	$O(d(E + R))$
TransH	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{w}_r \in \mathbb{R}^d$	$-\ \mathbf{h} - \mathbf{w}_r^T \mathbf{h} \mathbf{w}_r + \mathbf{r} - \mathbf{w}_r^T \mathbf{t} \mathbf{w}_r\ _2^2$	$\ \mathbf{h}\ _2 \leq 1, \ \mathbf{t}\ _2 \leq 1$ $ \mathbf{w}_r^T \mathbf{h} / \ \mathbf{r}\ _2 \leq \epsilon,$ $\ \mathbf{w}_r\ _2 \leq 1$	$O(d(E + 2 R))$
TransR	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^k,$ $\mathbf{M}_r \in \mathbb{R}^{k \times d}$	$-\ \mathbf{M}_r \mathbf{h} + \mathbf{r} - \mathbf{M}_r \mathbf{t}\ _2^2$	$\ \mathbf{h}\ _2 \leq 1, \ \mathbf{t}\ _2 \leq 1, \ \mathbf{r}\ _2 \leq 1$ $\ \mathbf{M}_r \mathbf{h}\ _2 \leq 1,$ $\ \mathbf{M}_r \mathbf{t}\ _2 \leq 1$	$O(d(E + k R))$

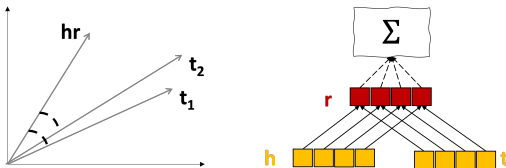
- ☞ Translation-based models translate the head entity embedding by summing with the relation embedding vector, measuring the distance (based on L_1 or L_2) between the translated images of head entity and the tail entity embedding, under strong assumptions about the relation embedding and the entities embedding into a relation-specific space before translation.
- ☞ TransE (on reflexive/one-to-many/many-to-one/many-to-many relations) was the first, many extensions such as TransR, TransH, and TransA . These models are generally simple and efficient.

Multi-relational Graph Embedding: DistMult

- Bilinear-product-based models compute their scores by using bilinear product between head, tail, and relation embeddings, DistMult the simplest one in this category [↗](#) and ComplEx,

$$f_r(h, t) = \sum_i \mathbf{h}_i \mathbf{r}_i \mathbf{t}_i$$

- The intuition behind is to compute the cosine similarity between $\mathbf{h} \times \mathbf{r}$ and $\mathbf{t}$



Multi-relational Graph Embedding: Neural-network-based

- These models use a nonlinear  neural network to compute the matching score for a triple:

$$f_r(h, t) = NN(\mathbf{h}, \mathbf{r}, \mathbf{t})$$

- One of the simplest *NN* neural-network-based model is ER-MLP which concatenates the input embedding vectors and uses a multi-layer perceptron neural network to compute the matching score.
- These models are complicated because of their use of neural networks as a black-box universal approximator, which usually make them difficult to understand and expensive to use.

Multi-relational Graph Embedding: Model performances

	Task	Example	Response
Link Prediction	Triple classification	(Joseph Fourier, occupation, physicist) ?	(yes, 95%)
	Tail classification	(Joseph Fourier, occupation, ?)	(1, physicist, 95%) (2, chemist, 93%)
	Head classification	(?, occupation, physicist)	(1, Albert Einstein, 91%), (2, Stephan Hawking, 89%)
	Relation prediction	(Joseph Fourier, ?, physicist)	(1, occupation, 95%)
	Entity classification	(Joseph Fourier, isTypeOf, ?)	(1, Person, 95%) (1, Scientist, 99%)

- Performance indicators are often used to measure the embedding quality of a model. Their simplicity makes them very suitable for evaluating the performance of an embedding algorithm even on a large scale. Given Q as the set of all ranked predictions of a model:
- Hits@ K or $H@K = \frac{|\{q \in Q: q < k\}|}{|Q|} \in [0, 1]$ measures the probability to find the correct prediction in the first top K model predictions. Usually, used for $k = 10$, it reflects the accuracy of an embedding model to predict the relation between two given triples correctly. Larger values mean better predictive performances.
- Mean rank is the average ranking position of the triples predicted correctly by the model among all the possible triples. $MR = \frac{1}{|Q|} \sum_{q \in Q} q$
The smaller the value, the better the model.
- Mean reciprocal rank measures the number of triples predicted correctly. $MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{q} \in [0, 1]$
The larger the value, the better the model.