Knowledge Graph Embedding

Invited Lecture (Graph7) at CMU 11441/11741

Presenter: Shanda Li

Slides credit to Zhiqing Sun (zhiqings@cs.cmu.edu)

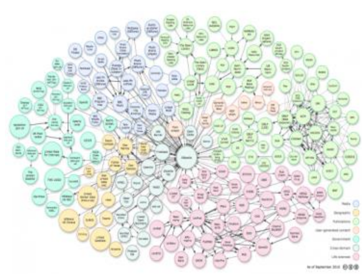
Carnegie Mellon University

Language Technologies Institute

1

Knowledge Graphs

- A set of facts represented as triplets
  - (head entity, relation, tail entity)
- A variety of applications
  - Question answering
  - Search
  - Recommender Systems
  - Natural language understanding
  - …



Google Knowledge Graph

Microsoft SATORI

NELL: Never-Ending Language Learning

Freebase DBpedia yago select knowledge

OpenIE (Reverb, OLLIE)

2

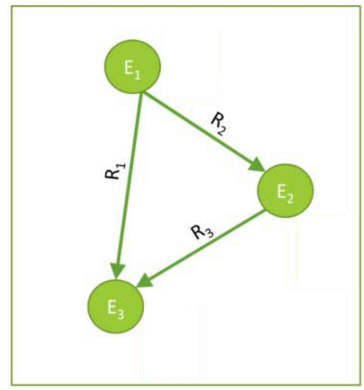
2

Knowledge Graphs: Knowledge in graph form

- A set of facts represented as triplets
  - with (h, r, t) for head entity, relation, tail entity

- Nodes are **entities**
- Nodes are labeled with their **types**
- Edges between two nodes capture **relationships** between entities

- KG is an example of a **heterogeneous** graph

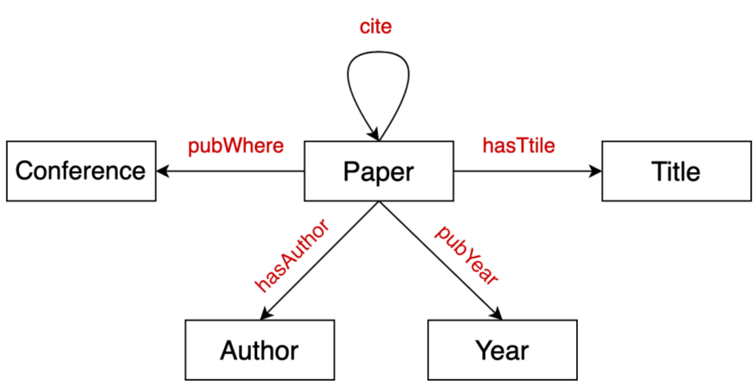


3

3

Example: Bibliographic Networks

- **Node types**: paper, title, author, conference, year
- **Relationships**: pubWhere, pubYear, hasTitle, hasAuthor, cite

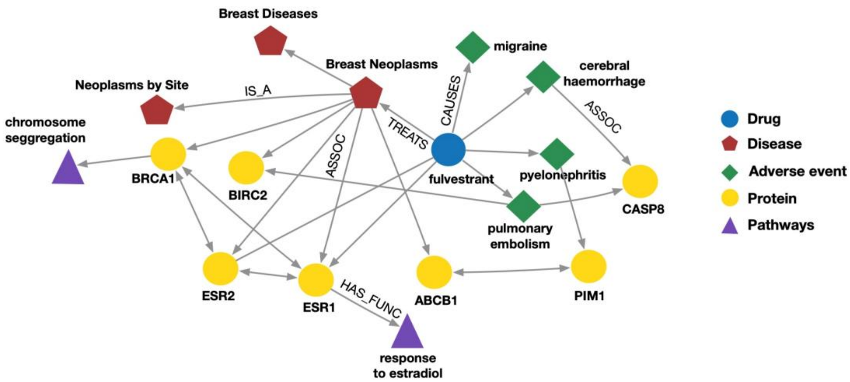


4

4

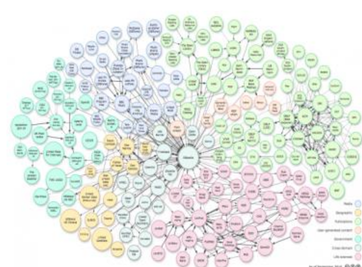
Example: Biological Knowledge Graphs

- **Node types**: drug, disease, adverse event, protein, pathways
- **Relationships**: has_func, causes, assoc, treats, is_a



Knowledge Graph Datasets

- Publicly available KGs:
  - FreeBase, Wikidata, Dbpedia, YAGO, NELL, etc.
- Common characteristics:
  - **Massive**: Millions of nodes and edges
  - **Incomplete**: Many true edges are missing
- Examples: Freebase
  - 93.8% of persons from Freebase have no place of birth and 78.5% have no nationality!



Task: Knowledge Graph Completion

- A fundamental task: **predicting missing links**
- The Key Idea: model and infer the **relation patterns** in knowledge graphs according to observed knowledge facts.
  - The relationship between relations

- Example:

Obama_Barack **Wife** Michelle_Obama

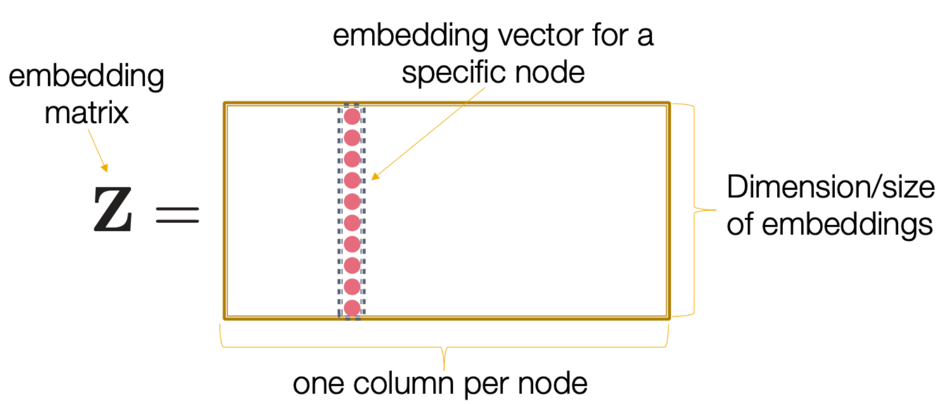
Michelle_Obama **Husband** Obama_Barack

Parents of Parents are Grandparents

7

Knowledge Graph Embedding

- Representing entities as **embeddings**



8

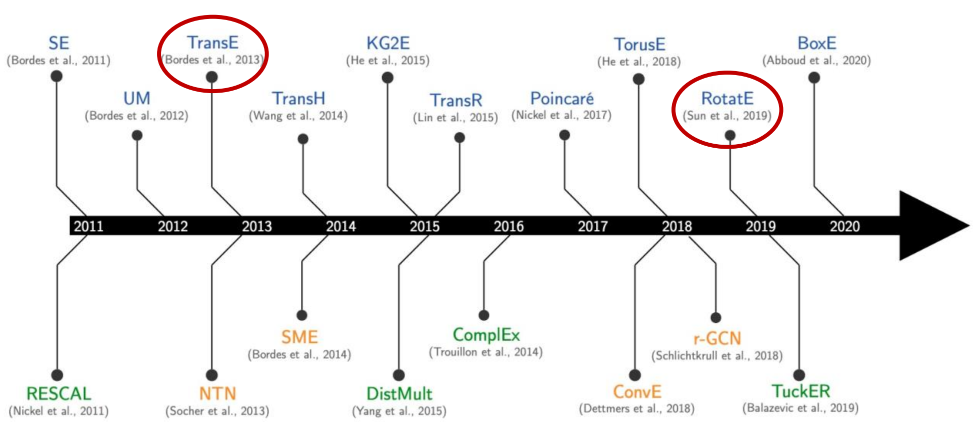
Knowledge Graph Embedding

- Representing entities as **vectors**
- Representing relations as **vectors** or **matrices**
- Denoting by (h, r, t) as a triplet of head (h), relation (r), and tail (t).
- For each semantically valid (h, r, t), we want to establish a mapping $f: (\boldsymbol{h}, \boldsymbol{r}) \rightarrow \boldsymbol{t}'$ such that $\boldsymbol{t}'$ is close to the true $\boldsymbol{t}$ where the boldfaced letters are the embeddings $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}, \boldsymbol{t}' \in \mathbb{R}^{\boldsymbol{d}}$, respectively.
- We define scoring function $\phi(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) \in \mathbb{R}$ which takes the embeddings $\boldsymbol{h}, \boldsymbol{r}$ *and* $\boldsymbol{t}$ as its input, and returns a high score if (h, r, t) is semantically valid, and a low score otherwise.

9

9

Related Work in Knowledge Graph Embedding



10

10

Desiderata for good knowledge graph embedding

- The embeddings need to be ***expressive*** enough to model typical relational patterns.
  - **Symmetric / Antisymmetric** Relations;
  - **Inverse** Relations;
  - **Composition** Relations;
  - Etc.

11

Relational Patterns

- **Symmetric vs. Antisymmetric** Relations
  - Symmetric: e.g., Marriage ("A is married B" means that "B is marred A").
  - Antisymmetric: e.g., Filiation ("A is a son of B" means that "B is not a son of A").
- Formally
  - For relation r to be symmetric, we have

$$r(x, y) \text{ is true} \Rightarrow r(y, x) \text{ is true}, \ \forall x, y$$

  - For relation r to be antisymmetric, we have

$$r(x, y) \text{ is true} \Rightarrow \neg r(y, x) \text{ is true}, \forall x, y$$

12

Relational Patterns

- **Inverse** Relations
  - E.g., "A is the husband of B" and "B is the wife of A"
- Formally
  - For two relations r_1 and r_2 to be inversely related, we have

$$r_2(x, y) \text{ is true} \Rightarrow r_1(y, x) \text{ is true}, \forall x, y$$

Relational Patterns

- **Composition** Relations
  - My mother's husband is my father (i.e., "A is the husband of B" and " B is the mother of C" implies that "A is the father of C")
- Formally
  - For relation r_3 to be a composition of r_1 and r_2, we have

$$r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z), \forall\, x, y, z$$

Abilities in Inferring the Relation Patterns

- None of existing methods can model and infer all the four types of relation patterns except RotatE

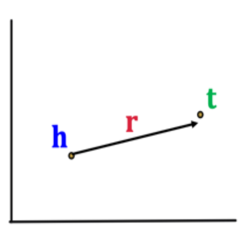
Model	Score Function	Symmetry	Antisymmetry	Inversion	Composition
SE	$-\|\boldsymbol{W}_{r,1}\mathbf{h} - \boldsymbol{W}_{r,2}\mathbf{t}\|$	✗	✗	✗	✗
TransE	$-\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$	✗	✓	✓	✓
TransX	$-\|g_{r,1}(\mathbf{h}) + \mathbf{r} - g_{r,2}(\mathbf{t})\|$	✓	✓	✗	✗
DistMult	$\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$	✓	✗	✗	✗
ComplEx	$\mathrm{Re}(\langle \mathbf{h}, \mathbf{r}, \overline{\mathbf{t}} \rangle)$	✓	✓	✓	✗
RotatE	$-\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|$	✓	✓	✓	✓

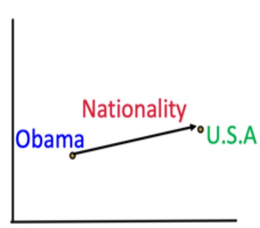
15

15

TransE (Bordes et al., NIPS 2013)

- Denote by boldfaced $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \in \mathbb{R}^{\boldsymbol{d}}$ the embeddings of the **h**ead, **r**elation and **t**ail in triplet (h, r, t), respectively.
- TransE's Objective: Find the embeddings such that **h** + **r** ≈ **t** if triplet (h, r, t) exists in the KG, and otherwise **h** + **r** ≠ **t.**
- Scoring Function $\phi(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = -\|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|$





16

16

Supervised Learning of Embeddings

- Training set $S = S^+ \cup S^-$
  - $S^+ \coloneqq \{sampled\ triplets\ (h,r,t) \in KG\}$ as the possible instances
  - $S^- \coloneqq \{(h',r,t) \cup (h,r,t') \notin KG\}$ as the negative instances
- Scoring function (higher is better) for any triplets

$$\phi(h,r,t) = -\|h + r - t\| = -distance(t,\hat{t})$$

- Optimize entity/relation embeddings as

$$\max_{(h,r,t)} \sum_{x \in S^+} \sum_{x' \in S^-} [\phi(x) - \phi(x') - \gamma]_+$$

where x and x' are two triplets; γ is a margin hyperparameter; $[.]_+$ is the hinge loss.

17

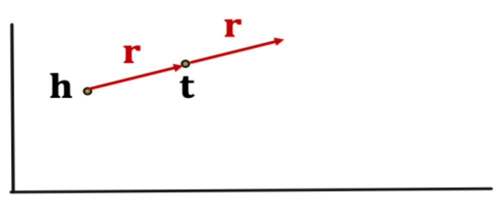
17

Analysis of TransE (Bordes et al., NIPS 2013)

- **Antisymmetric Relations:**

$$r(h,t) \Rightarrow \neg r(t,h) \quad \forall h,t$$

  - **Example**: Hypernym (a word with a broader meaning: dog v.s. poodle)
- **TransE** can model antisymmetric relations ✓
  - $\mathbf{h} + \mathbf{r} = \mathbf{t}$, but $\mathbf{t} + \mathbf{r} \neq \mathbf{h}$



18

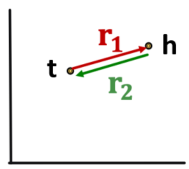
18

Analysis of TransE (Bordes et al., NIPS 2013)

- **Inverse Relations:**

  $$r_2(h,t) \Rightarrow r_1(t,h)$$

  - **Example** : (Advisor, Advisee)
- TransE can model inverse relations ✓
  - $\mathbf{h} + \mathbf{r}_2 = \mathbf{t}$, we can set $\mathbf{r}_1 = -\mathbf{r}_2$



19

19

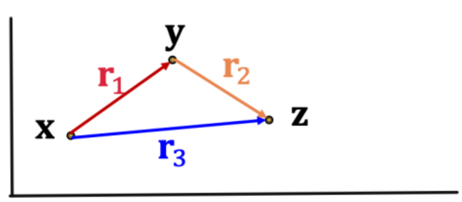
Analysis of TransE (Bordes et al., NIPS 2013)

- **Composition (Transitive) Relations:**

  $$r_1(x,y) \wedge r_2(y,z) \Rightarrow r_3(x,z) \quad \forall x,y,z$$

  - **Example**: My mother's husband is my father.
- TransE can model composition relations ✓

  $$\mathbf{r}_3 = \mathbf{r}_1 + \mathbf{r}_2$$



20

20

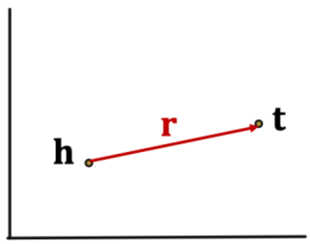
Analysis of TransE (Bordes et al., NIPS 2013)

- **Symmetric Relations:**

$$r(h,t) \Rightarrow r(t,h) \quad \forall h,t$$

  - **Example**: Family, Roommate
- **TransE cannot** model symmetric relations ✗

  only if $\mathbf{r} = 0$, $\mathbf{h} = \mathbf{t}$



For all h, t that satisfy $r(h,t)$, $r(t,h)$ is also True, which means $||\mathbf{h} + \mathbf{r} - \mathbf{t}|| = 0$ and $||\mathbf{t} + \mathbf{r} - \mathbf{h}|| = 0$. Then $\mathbf{r} = 0$ and $\mathbf{h} = \mathbf{t}$, however h and t are two different entities and should be mapped to different locations.

21

21

RotatE (Sun et al., ICLR'2019)

- RotatE treats each relation as an operator of **elementwise rotation** from the source entity vector to the target entity vector in the **complex** vector space.
- RotatE can model and infer all the four types of relation patterns.
- RotatE offers an efficient and effective negative sampling algorithm for optimization.
- RotatE achieved SOTA results (at the time) on all the evaluation benchmarks for link prediction over knowledge graphs

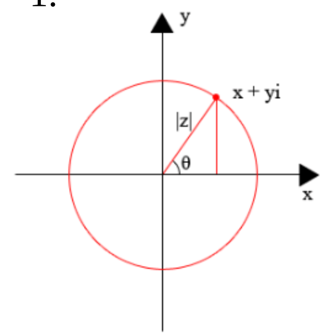
Zhiqing Sun, Zhihong Deng, Jian-Yun Nie, and Jian Tang. "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space." ICLR'19.

22

22

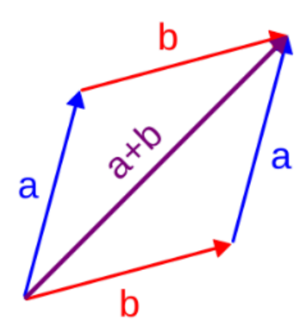
Complex number refresher

- The complex plane $\mathbf{C}=\{x+\mathrm{i}y \mid x, y\in\mathbf{R}\}$, where $\mathrm{i}^2=-1$.
- Geometric representation of $z\in\mathbf{C}$:
  - A point/vector (x, y) in the 2D plane.
  - *Modulus* or *absolute value*: $\rho=|z|$
  - *Argument*: θ
- Polar coordinate representation of $z\in\mathbf{C}$:
  - $z=\rho\cos\theta+\mathrm{i}\,\rho\sin\theta=\rho e^{\mathrm{i}\theta}$.
  - Euler's formula: $e^{\mathrm{i}\theta}=\cos\theta+\mathrm{i}\sin\theta$.



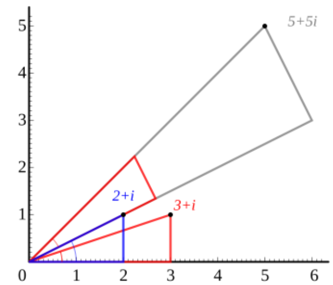
Complex addition

- Mathematical formula:
  - $(x_1+\mathrm{i}y_1)+(x_2+\mathrm{i}y_2)=(x_1+x_2)+\mathrm{i}(y_1+y_2)$
- Geometric interpretation:
  - Vector addition in the 2D plane.
  - Parallelogram/Triangle law.



Complex multiplication

- Mathematical formula:
  - $(x_1+iy_1)(x_2+iy_2)=(x_1x_2-y_1y_2)+i(x_1y_2+x_2y_1)$
  - $(\rho_1 e^{i\alpha})\cdot(\rho_2 e^{i\beta})=\rho_1\rho_2 e^{i\alpha+\beta}$



- Geometric interpretation:
  - Multiplying $\rho_1 e^{i\alpha}$ with $\rho_2 e^{i\beta}$ is equivalent to
  - (1) rescale the modulus of $\rho_1 e^{i\alpha}$ by a factor of ρ_2; and
  - (2) rotate the argument of $\rho_1 e^{i\alpha}$ by β.

- A special case: Multiplying with $e^{i\beta}$ = Rotation by β.

25

Relation as Elementwise Rotation

- Define head and tail entities as $\mathbf{h}, \mathbf{t} \in \mathbb{C}^{\boldsymbol{k}}$ in a complex vector space.
- Define relation $\mathbf{r}$ as an element-wise rotation operator from head $\mathbf{h}$ to tail $\mathbf{t}$ such that

$$t_j = h_j r_j \text{ where } |r_j|=1 \text{ for } j \in \{1, \dots, k\}.$$

- The relation $\mathbf{r}$ is parametrized using the arguments (i.e., rotation angles) of its entries:

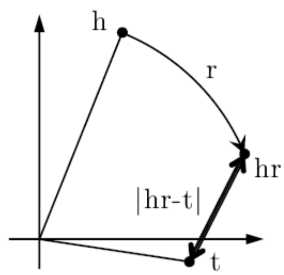
$$\boldsymbol{\theta}_r = (\theta_{r,1}, \theta_{r,2}, \dots, \theta_{r,k}); \mathbf{r} = (e^{i\theta_{r,1}}, e^{i\theta_{r,2}}, \dots, e^{i\theta_{r,k}}).$$

26

Geometric Interpretation

- Define the distance function of RotatE as

$$d_{\boldsymbol{r}}(\boldsymbol{h}, \boldsymbol{t}) = ||\mathbf{h} \circ \mathbf{r} - \mathbf{t}|| \quad (\circ \text{ is the Hadamard product})$$



27

27

Modeling Symmetry and Antisymmetry

$r_i = e^{i\theta_{r,i}}$
$d_{\boldsymbol{r}}(\boldsymbol{h}, \boldsymbol{t}) = ||\mathbf{h} \circ \mathbf{r} - \mathbf{t}||$

A relation **r** is **symmetric** if $r(h,t) \Rightarrow r(t,h) \quad \forall h,t$

➢In RotatE, this is equivalent to $\boldsymbol{t} = \boldsymbol{h} \circ \boldsymbol{r} \Rightarrow \boldsymbol{h} = \boldsymbol{t} \circ \boldsymbol{r}$.

➢Eliminating t yields $\boldsymbol{h} = \boldsymbol{h} \circ \boldsymbol{r} \circ \boldsymbol{r}$ $(\forall\, \boldsymbol{h})$.

28

28

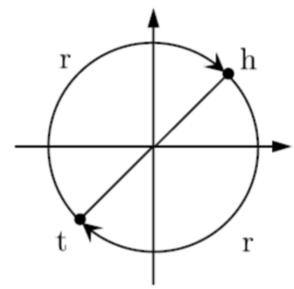
Modeling Symmetry and Antisymmetry

$r_i = e^{i\theta_{r,i}}$
$d_r(\boldsymbol{h}, \boldsymbol{t}) = ||\mathbf{h} \circ \mathbf{r} - \mathbf{t}||$

A relation **r** is **symmetric** if $r(h, t) \Rightarrow r(t, h) \quad \forall h, t$

➢In RotatE, this is equivalent to $\boldsymbol{t} = \boldsymbol{h} \circ \boldsymbol{r} \Rightarrow \boldsymbol{h} = \boldsymbol{t} \circ \boldsymbol{r}$.

➢Eliminating t yields $\boldsymbol{h} = \boldsymbol{h} \circ \boldsymbol{r} \circ \boldsymbol{r}$ $(\forall\, \boldsymbol{h})$.



[Conclusion] A relation $\boldsymbol{r}$ is **symmetric** if and only if $\boldsymbol{r} \circ \boldsymbol{r} = \mathbf{1}$, or equivalently $r_j^2 = 1$, i.e.,

$$\theta_{r,j} \in \{0, \pi\}\ (\forall j \in \{1, \dots, k\}).$$

A relation $\boldsymbol{r}$ is **antisymmetric** if and only if $\boldsymbol{r} \circ \boldsymbol{r} \neq \mathbf{1}$.

29

29

Modeling Inverse Relations

$r_i = e^{i\theta_{r,i}}$
$d_r(\boldsymbol{h}, \boldsymbol{t}) = ||\mathbf{h} \circ \mathbf{r} - \mathbf{t}||$

Two relations $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are **inverse** if $\boldsymbol{r_1}(\boldsymbol{h}, \boldsymbol{t}) \Rightarrow \boldsymbol{r_2}(\boldsymbol{t}, \boldsymbol{h})$.

➢In RotatE, this is equivalent to $\boldsymbol{t} = \boldsymbol{h} \circ \boldsymbol{r_1} \Rightarrow \boldsymbol{h} = \boldsymbol{t} \circ \boldsymbol{r_2}$.

➢Eliminating t yields $\boldsymbol{h} = \boldsymbol{h} \circ \boldsymbol{r_1} \circ \boldsymbol{r_2}$ $(\forall\, \boldsymbol{h})$.

30

30

Modeling Inverse Relations

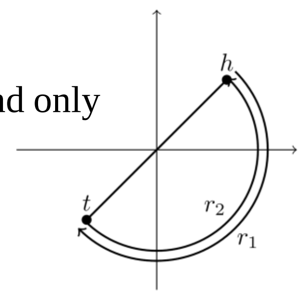
$r_i = e^{i\theta_{r,i}}$

$d_r(\boldsymbol{h}, \boldsymbol{t}) = ||\mathbf{h} \circ \mathbf{r} - \mathbf{t}||$

Two relations $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are **inverse** if $\boldsymbol{r_1}(\boldsymbol{h}, \boldsymbol{t}) \Rightarrow \boldsymbol{r_2}(\boldsymbol{t}, \boldsymbol{h})$.

➢In RotatE, this is equivalent to $\boldsymbol{t} = \boldsymbol{h} \circ \boldsymbol{r_1} \Rightarrow \boldsymbol{h} = \boldsymbol{t} \circ \boldsymbol{r_2}$.

➢Eliminating t yields $\boldsymbol{h} = \boldsymbol{h} \circ \boldsymbol{r_1} \circ \boldsymbol{r_2}$ ($\forall\ \boldsymbol{h}$).



[Conclusion] Two relations $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are **inverse** if and only if $\boldsymbol{r_1} \circ \boldsymbol{r_2} = \mathbf{1}$, or equivalently $r_{1j} r_{2j} = 1$, i.e.,

$$\theta_{r_1,j} + \theta_{r_2,j} = 0\ (\forall j \in \{1, \dots, k\}).$$

31

31

Modeling Compositional Relations

$r_i = e^{i\theta_{r,i}}$

$d_r(\boldsymbol{h}, \boldsymbol{t}) = ||\mathbf{h} \circ \mathbf{r} - \mathbf{t}||$

A relation $\boldsymbol{r_3}$ is a **composition** of relations $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ if

$$\boldsymbol{r_1}(\boldsymbol{x}, \boldsymbol{y}) \wedge \boldsymbol{r_2}(\boldsymbol{y}, \boldsymbol{z}) \Rightarrow \boldsymbol{r_3}(\boldsymbol{x}, \boldsymbol{z}).$$

➢In RotatE, this is equivalent to

$$\boldsymbol{y} = \boldsymbol{x} \circ \boldsymbol{r_1} \wedge \boldsymbol{z} = \boldsymbol{y} \circ \boldsymbol{r_2} \Rightarrow \boldsymbol{z} = \boldsymbol{x} \circ \boldsymbol{r_3}.$$

➢Eliminating $\boldsymbol{y}$ yields $\boldsymbol{x} \circ \boldsymbol{r_3} = \boldsymbol{x} \circ \boldsymbol{r_1} \circ \boldsymbol{r_2}$ ($\forall\ \boldsymbol{x}$).

32

32

Modeling Inverse Relations

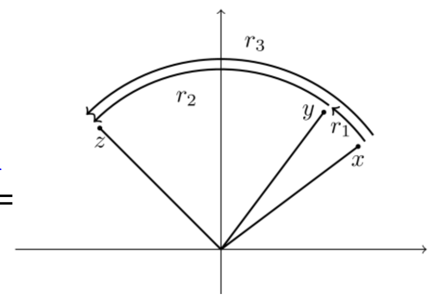
$r_i = e^{i\theta_{r,i}}$
$d_r(\boldsymbol{h}, \boldsymbol{t}) = ||\mathbf{h} \circ \mathbf{r} - \mathbf{t}||$

A relation $\boldsymbol{r_3}$ is a **composition** of relations $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ in RotatE if

$$\boldsymbol{x} \circ \boldsymbol{r_3} = \boldsymbol{x} \circ \boldsymbol{r_1} \circ \boldsymbol{r_2} \; (\forall \, \boldsymbol{x}).$$

[Conclusion] A relation $\boldsymbol{r_3}$ is a **composition** of relations $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ if and only if $\boldsymbol{r_1} \circ \boldsymbol{r_2} = \boldsymbol{r_3}$, or equivalently

$$\theta_{r_1,j} + \theta_{r_2,j} = \theta_{r_3,j} = (\forall j \in \{1, \ldots, k\}).$$



33

Optimization

- Negative sampling loss

Positive instance   Negative instance

$$L = -\log \sigma\big(\gamma - d_r(\boldsymbol{h}, \boldsymbol{t})\big) - \mathbf{E}_{(\boldsymbol{h}', r, \boldsymbol{t}')}[\log \sigma(d_r(\boldsymbol{h}', \boldsymbol{t}') - \gamma)]$$

- γ is a fixed margin
- σ is the sigmoid function

34

Self-adversarial Negative Sampling

- Traditionally, the negative samples are drawn in a uniform way
  - Inefficient as training goes on since many samples are obviously false
  - Does not provide useful information
- A self-adversarial negative sampling
  - Sample negative triplets according to the current embedding model
  - Starts from easier samples to more and more difficult samples
  - Curriculum Learning

$$p(h'_j, r, t'_j | \{(h_i, r_i, t_i)\}) = \frac{\exp \alpha f_r(\mathbf{h}'_j, \mathbf{t}'_j)}{\sum_i \exp \alpha f_r(\mathbf{h}'_i, \mathbf{t}'_i)}$$

- α is the temperature of sampling. $f_r(h'_j, t'_j)$ measures the salience of the triplet

Experimental Results on FB15K and WN18

- Task: (h, r, ?) or (?, r, t)
- RotatE achieves state-of-the-art performance

	FB15k					**WN18**				
	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
TransE [♥]	-	.463	.297	.578	.749	-	.495	.113	.888	.943
DistMult [♦]	42	.798	-	-	**.893**	655	.797	-	-	.946
HolE	-	.524	.402	.613	.739	-	.938	.930	.945	.949
ComplEx	-	.692	.599	.759	.840	-	.941	.936	.945	.947
ConvE	51	.657	.558	.723	.831	374	.943	.935	.946	.956
pRotatE	43	**.799**	**.750**	.829	.884	**254**	.947	.942	.950	.957
RotatE	**40**	.797	.746	**.830**	.884	309	**.949**	**.944**	**.952**	**.959**

Results on FB15k-237 and WN18RR

- RotatE achieves state-of-the-art performance

	FB15k-237					WN18RR				
	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
TransE [♥]	357	.294	-	-	.465	3384	.226	-	-	.501
DistMult	254	.241	.155	.263	.419	5110	.43	.39	.44	.49
ComplEx	339	.247	.158	.275	.428	5261	.44	.41	.46	.51
ConvE	244	.325	.237	.356	.501	4187	.43	.40	.44	.52
pRotatE	178	.328	.230	.365	.524	**2923**	.462	.417	.479	.552
RotatE	**177**	**.338**	**.241**	**.375**	**.533**	3340	**.476**	**.428**	**.492**	**.571**

Further Experiments

- Comparing with the other models trained with self-adversarial
- Similar results are observed

	FB15k		FB15k-237	
	MRR	H@10	MRR	H@10
TransE	.735	.871	.332	.531
ComplEx	.780	**.890**	.319	.509
RotatE	**.797**	.884	**.338**	**.533**

Summary

- Modeling relation patterns is critical for knowledge base completion
  - Symmetric/Antisymmetric, Inverse, and composition
- RotatE: define each relation as an **element-wise rotation** from the head entity to the tail entity in the complex vector space
  - Capable of modeling and inferring all the four types of relation patterns
- A new self-adversarial negative sampling approach
  - Sampling the negative samples according to current embeddings
  - Curriculum learning
- State-of-the-art results on all existing benchmark datasets

39