Analogical Inference for Multi-Relational Embeddings

Hanxiao Liu, Yuexin Wu, Yiming Yang
Carnegie Mellon University

August 8, 2017
Task Description

Multi-Relational Embeddings:
- Finding latent representations of entities and relations.
- Useful for knowledge base completion (by discovering missing facts), etc.

Novel Contribution:
- Instead of tradition rule-based AI, we impose *analogical structures* in the learning of entity/relation embedding.
Why Analogy? (a toy example)

**Figure:** Solar System (blue) v.s. Atomic System (red).

Knowing the relational structure in one system will help us to understand the other system by analogy.
Basic Formulation

- Denote by vector $v_e$ the embedding of entity $e$.
- Denote by matrix $W_r$ in the embedding of relation $r$.
- Assume all valid subject-relation-object $(s,r,o)$ triples approximately satisfy

$$v_s^\top W_r \approx v_o^\top \quad (1)$$

- Define the scoring function of any $(s,r,o)$ triple as:

$$\phi(s, r, o) = \langle v_s^\top W_r, v_o \rangle = v_s^\top W_r v_o \quad (2)$$
The family of matrices satisfying:

\[ W_r^\top W_r = W_r W_r^\top \]  

(3)

Special cases:

1. **Symmetric Matrices**
   - \( \phi(s, r, o) = \phi(o, r, s) \). E.g. *is_identical*.

2. **Skew-symmetric Matrices**
   - \( \phi(s, r, o) = -\phi(o, r, s) \). E.g. *is_parent_of*.

3. **Orthogonal Matrices**
   - Useful if \( r \) is a bijection (one-to-one mapping).
Observation: Analogical structures often imply “parallelograms”, e.g.,

“man is to king as woman is to queen”

Or, in an abstract notion:

“a is to b as c is to d”

Given the parallelogram, if we know $a \xrightarrow{r} b$ and $a \xrightarrow{r'} c$, then $c \xrightarrow{r} d$ and $b \xrightarrow{r'} d$ can be inferred by symmetry.
Mathematically, the necessary condition for having an analogical structure is the commutativity of relations:

\[ r \circ r' = r' \circ r \quad (4) \]

Equivalently, we want the following constraint:

\[ W_r W_{r'} = W_{r'} W_r \quad (5) \]

![Diagram showing commutativity of relations in a graph with nodes a, b, c, and d and edges r, r', and r.](image-url)
Optimization: Straightforward Formulation

Notation: Label $y = +1$ for positive examples and $-1$ otherwise; Data distribution $\mathcal{D}$; Loss function $\ell$.

$$\min_{\mathbf{v}, \mathbf{W}} \mathbb{E}_{s, r, o, y \sim \mathcal{D}} \ell (\phi_{\mathbf{v}, \mathbf{W}}(s, r, o), y)$$  \hspace{1cm} (6)

s.t.  \hspace{0.5cm} W_r W_r^\top = W_r^\top W_r \hspace{0.5cm} \forall r \hspace{1cm} (7)

\hspace{0.5cm} W_r W_{r'} = W_{r'} W_r \hspace{0.5cm} \forall r, r' \hspace{1cm} (8)

- (7) follows the definition of normal matrices.
- (8) is for the communicative property.

The OPT is expensive due to (i) $W_r$’s are fully dense matrices (ii) large number of equality constraints.
Solution $\mathbf{v}^*, \mathbf{W}^*$ for the previous OPT can be exactly recovered by solution $\mathbf{v}'^*, \mathbf{W}'^*$ of the following problem:

$$
\min_{\mathbf{v}', \mathbf{W}'} \mathbb{E}_{s,r,o,y \sim D} \ell (\phi_{\mathbf{v}'}, \mathbf{W}' (s, r, o), y)
$$

(9)

Most notably,

- We show that any $W'_r$ must be block-diagonal with the diagonal block sizes bounded by 2.
  - $O(m)$ free parameters in the $m \times m$ matrix.
- We now have an unconstrained optimization instead.
  - Efficiently solved using SGD without projection.
A Unified View of Existing Work

We explain the strong empirical performance of

- DistMult (Yang et al., ICLR 2015)
- ComplEx (Trouillon et al., ICML 2016)
- HolE (Nickel et al., AAAI 2016)

by showing that they are implicitly imposing analogical structures and are restricted cases of ours.
Connections to Existing Work

Multiplicative Embeddings (DistMult)

\[ \phi(s, r, o) = \langle v_s, v_r, v_o \rangle \]  \hspace{1cm} (10)

where \( v_s, v_r, v_o \in \mathbb{R}^m, \forall s, r, o \) \hspace{1cm} (11)

DistMult embeddings of size \( m \) can be fully recovered by ANALOGY embeddings of size \( m \).

Intuition: \( v_r \) can be viewed as a diagonal \( W_r \overset{def}{=} \text{diag}(v_r) \).
Diagonal matrices are always commutative.
Complex Embeddings (ComplEx)

\[ \phi(s, r, o) = \Re(\langle v_s, v_r, v_o \rangle) \]  \hspace{1cm} (12)

where \( v_s, v_r, v_o \in \mathbb{C}^m, \forall s, r, o \)  \hspace{1cm} (13)

ComplEx embeddings of size \( m \) can be fully recovered by ANALOGY embeddings of size \( 2m \).

Intuition: there exists a bijection between any \( a + bj \in \mathbb{C} \) and \( \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \in \mathbb{R}^{2\times2} \).
Holographic Embeddings (HolE)

\[ \phi(s, r, o) = \langle v_r, v_s \ast v_o \rangle \quad (14) \]

where \( v_s, v_r, v_o \in \mathbb{R}^m, \forall s, r, o \) \hfill (15)

HolE embeddings can be equivalently obtained via

\[ \phi(s, r, o) = \Re (\langle v_s, v_r, \overline{v_o} \rangle) \quad (16) \]

where \( v_s, v_r, v_o \in \text{FFT}(\mathbb{R}^m) \in \mathbb{C}^m, \forall s, r, o \) \hfill (17)

Hence is a restricted case of ComplEx and ANALOGY.

Intuition: Circular convolution \( \ast \) can be converted into element-wise product after Fourier transform.
Experiments

Implementation Details

- Use logistic loss:

\[ \ell(\phi(s, r, o), y) = - \log \sigma(y\phi(s, r, o)) \]  \hspace{1cm} (18)

- Optimization: Asynchronous AdaGrad (HogWild!)

- For each valid \((s, r, o)\), generate negative examples \((s', r, o)\), \((s, r', o)\), \((s, r, o')\) by corrupting \(s, r, o\).

Evaluation

- Hits and Mean Reciprocal Rank (MRR)
## Results – Hits@10 (filt.)

<table>
<thead>
<tr>
<th>Models</th>
<th>WN18</th>
<th>FB15K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unstructured</td>
<td>38.2</td>
<td>6.3</td>
</tr>
<tr>
<td>RESCAL</td>
<td>52.8</td>
<td>44.1</td>
</tr>
<tr>
<td>NTN</td>
<td>66.1</td>
<td>41.4</td>
</tr>
<tr>
<td>SME</td>
<td>74.1</td>
<td>41.3</td>
</tr>
<tr>
<td>SE</td>
<td>80.5</td>
<td>39.8</td>
</tr>
<tr>
<td>LFM</td>
<td>81.6</td>
<td>33.1</td>
</tr>
<tr>
<td>TransH</td>
<td>86.7</td>
<td>64.4</td>
</tr>
<tr>
<td>TransE</td>
<td>89.2</td>
<td>47.1</td>
</tr>
<tr>
<td>TransR</td>
<td>92.0</td>
<td>68.7</td>
</tr>
<tr>
<td>TKRL</td>
<td>–</td>
<td>73.4</td>
</tr>
<tr>
<td>RTransE</td>
<td>–</td>
<td>76.2</td>
</tr>
<tr>
<td>TransD</td>
<td>92.2</td>
<td>77.3</td>
</tr>
<tr>
<td>CTransR</td>
<td>92.3</td>
<td>70.2</td>
</tr>
<tr>
<td>KG2E</td>
<td>93.2</td>
<td>74.0</td>
</tr>
<tr>
<td>STransE</td>
<td>93.4</td>
<td>79.7</td>
</tr>
<tr>
<td>DistMult</td>
<td>93.6</td>
<td>82.4</td>
</tr>
<tr>
<td>TransSparse</td>
<td>93.9</td>
<td>78.3</td>
</tr>
<tr>
<td>PTransE-MUL</td>
<td>–</td>
<td>77.7</td>
</tr>
<tr>
<td>PTransE-RNN</td>
<td>–</td>
<td>82.2</td>
</tr>
<tr>
<td>PTransE-ADD</td>
<td>–</td>
<td>84.6</td>
</tr>
<tr>
<td>NLF (+external data)</td>
<td>94.3</td>
<td>87.0</td>
</tr>
<tr>
<td>ComplEx</td>
<td>94.7</td>
<td>84.0</td>
</tr>
<tr>
<td>HolE</td>
<td>94.9</td>
<td>73.9</td>
</tr>
<tr>
<td><strong>Our ANALOGY</strong></td>
<td><strong>94.7</strong></td>
<td><strong>85.4</strong></td>
</tr>
</tbody>
</table>
## Results – Hits@\{1,3\} & MRR

<table>
<thead>
<tr>
<th>Models</th>
<th>MRR (filt.)</th>
<th>MRR (raw)</th>
<th>Hits@1 (filt.)</th>
<th>Hits@3 (filt.)</th>
<th>MRR (filt.)</th>
<th>MRR (raw)</th>
<th>Hits@1 (filt.)</th>
<th>Hits@3 (filt.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESCAL</td>
<td>89.0</td>
<td>60.3</td>
<td>84.2</td>
<td>90.4</td>
<td>35.4</td>
<td>18.9</td>
<td>23.5</td>
<td>40.9</td>
</tr>
<tr>
<td>TransE</td>
<td>45.4</td>
<td>33.5</td>
<td>8.9</td>
<td>82.3</td>
<td>38.0</td>
<td>22.1</td>
<td>23.1</td>
<td>47.2</td>
</tr>
<tr>
<td>DistMult</td>
<td>82.2</td>
<td>53.2</td>
<td>72.8</td>
<td>91.4</td>
<td>65.4</td>
<td>24.2</td>
<td>54.6</td>
<td>73.3</td>
</tr>
<tr>
<td>HolE</td>
<td>93.8</td>
<td>61.6</td>
<td>93.0</td>
<td>94.5</td>
<td>52.4</td>
<td>23.2</td>
<td>40.2</td>
<td>61.3</td>
</tr>
<tr>
<td>ComplEx</td>
<td>94.1</td>
<td>58.7</td>
<td>93.6</td>
<td>94.5</td>
<td>69.2</td>
<td>24.2</td>
<td>59.9</td>
<td>75.9</td>
</tr>
<tr>
<td><strong>Our ANALOGY</strong></td>
<td><strong>94.2</strong></td>
<td><strong>65.7</strong></td>
<td><strong>93.9</strong></td>
<td><strong>94.4</strong></td>
<td><strong>72.5</strong></td>
<td><strong>25.3</strong></td>
<td><strong>64.6</strong></td>
<td><strong>78.5</strong></td>
</tr>
</tbody>
</table>
Scalability

The algorithm scales linearly over the embedding size.

Figure: CPU run time per epoch (secs) of ANALOGY.

Intuition: $O(m)$ for almost-diagonal matrices instead of $O(m^2)$ for dense matrices.
Conclusion

Contributions:

- A new framework that \textit{explicitly} exploit analogy in a differentiable manner.
- Fast algorithm of linear scalability.
- Unified view of several representative works.

Future work: Other applications where analogies might be useful (Machine Translation, Image Captioning, etc.).
Poster #51

Code: https://github.com/quark0/ANALOGY

Thank You!