Typed Tensor Decomposition of Knowledge Bases for Relation Extraction

Kai-Wei Chang\(^1\), Wen-tau Yih\(^2\), Bishan Yang\(^3\), and Christopher Meek\(^2\)

\(^1\)University of Illinois, Urbana
\(^2\)Cornell University, Ithaca
\(^3\)Microsoft Research, Redmond

Presented by Yu Shi
Background

- $n$ entities $e_i$
- $m$ relations $r_k$
- entity-relation triples $(e_i, r_k, e_j)$

\[ \mathcal{T} = \{ (e_i, r_k, e_j) \} \]
Background

- Encode Binary Relations in a Tensor

\[ T = \{ (e_i, r_k, e_j) \} \]

\[ x_{i,j,k} = 1 \]

\[ \chi \in \{0, 1\}^{n \times n \times m} \]
Background

- Tensor Decomposition (RESCAL)

$x_k' \approx A R_k A^T$

Figure credits to Bordes, A. et al, Constructing and Mining Web-scale Knowledge Graphs, KDD2014
Background

• Tensor Decomposition (RESCAL)

\[ \mathcal{X}_k \approx \mathbf{A} \mathbf{R}_k \mathbf{A}^T \]

\[ f(\mathbf{A}, \mathbf{R}_k) = \frac{1}{2} \left( \sum_k \| \mathcal{X}_k - \mathbf{A} \mathbf{R}_k \mathbf{A}^T \|_F^2 \right) \]

\[ g(\mathbf{A}, \mathbf{R}_k) = \frac{1}{2} \left( \| \mathbf{A} \|_F^2 + \sum_k \| \mathbf{R}_k \|_F^2 \right) \]

\[ \min_{\mathbf{A}, \mathbf{R}_k} f(\mathbf{A}, \mathbf{R}_k) + \lambda \cdot g(\mathbf{A}, \mathbf{R}_k) \]

Figure credits to Bordes, A. et al, Constructing and Mining Web-scale Knowledge Graphs, KDD2014
Typed Tensor Decomposition

- Incorporate type information

- (Type constrains) For the $k$-th relation $(e_i, r_k, e_j)$ is a feasible triple iff $e_i \in \mathbb{L}_k$ and $e_j \in \mathbb{R}_k$

- Inference w.r.t. the $k$-th relation only involves submatrices $A_{k_l}$ and $A_{k_r}$
Typed Tensor Decomposition

- Incorporate type information

\[ f'(A, R_k) = \frac{1}{2} \sum_k \| \chi_{klr} - A_{kl} R_k A_{kr}^T \|_F^2 \]
Handling Regularization Efficiently

- Minimizing loss function by alternating least-squares method

\[
\mathbf{A} \leftarrow \left( \sum_k \left( \mathbf{x}_{k_lr} \mathbf{a}_{kr} \mathbf{R}_k^T + \mathbf{x}_{k_{lr}}^T \mathbf{a}_{k_l} \mathbf{R}_k \right) \right) \times \left[ \sum_k \mathbf{b}_{kr} + \mathbf{c}_{kl} + \lambda \mathbf{I} \right]^{-1}
\]

\[
\text{vec}(\mathbf{R}_k) \leftarrow \left( \mathbf{A}_{k_{lr}}^T \mathbf{A}_{kr} \otimes \mathbf{A}_{k_l}^T \mathbf{a}_{k_l} + \lambda \mathbf{I} \right)^{-1} \times \text{vec}(\mathbf{A}_{k_{lr}}^T \mathbf{x}_{k_{lr}} \mathbf{A}_{kr}),
\]

Inversion of an \( r^2 \times r^2 \) matrix
Handling Regularization Efficiently

\[ (A_k^T A \otimes A_k^T A + \lambda I)^{-1} \]

- Exploit the structure of tensor product and inversion
  \[ Z = A \otimes A \]

  \[
  (Z^T Z + \lambda I)^{-1} \]

  \[
  = (\lambda I + V \Sigma^2 V^T \otimes V \Sigma^2 V^T)^{-1} 
  = (\lambda I + (V \otimes V)(\Sigma^2 \otimes \Sigma^2)(V \otimes V)^T)^{-1} 
  = (V \otimes V)(\lambda I + \Sigma^2 \otimes \Sigma^2)^{-1}(V \otimes V)^T.
  \]

- Invert a diagonal matrix instead

\[ O(r^6 + pr^2) \]

\[ O(nr^2 + pr) \]
Experiments

- Outperforms previous methods in two tasks in knowledge base completion
  - **Entity** Retrieval
  - **Relation** Retrieval
Experiments

• Entity Retrieval

• Given a set of entity-relation pairs \( \{(e_i, r_k)\} \)

• Predict \( e_j \) such that \( (e_i, r_k, e_j) \in T \)
Experiments

• Relation Retrieval

  • Given a relation type $r_k$

  • Look for entity pairs $(e_i, e_j)$ that have this relationship, i.e., $(e_i, r_k, e_j) \in \mathcal{T}$
Experiments: Initialization

\[ \min_{A, R_k} f(A, R_k) + \lambda \cdot g(A, R_k) \]

- Not convex
- Initial A by SVD

\[ \bar{X} = \sum_k (x_k + x_k^T) \]
\[ \bar{X} = U \Sigma V^T \]

\[ A = U \]
Experiments: Baselines

- Two baseline methods
  - RESCAL
  - TransE
Experiments: Baselines

- RESCAL: tensor decomposition without type information

Figure credits to Bordes, A. et al, Constructing and Mining Web-scale Knowledge Graphs, KDD2014
Experiments: Baselines

- TransE: Translating Embeddings for Modeling Multi-relational Data

\[ -\|e_i + r_k - e_j\| \]

Figure credits to Wang, Z. et al, Knowledge Graph Embedding by Translating on Hyperplanes, AAAI2014
Experiments: Results

<table>
<thead>
<tr>
<th>w/o type checking</th>
<th>Entity Retrieval</th>
<th>Relation Retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TransE</td>
<td>RESCAL</td>
</tr>
<tr>
<td></td>
<td>51.41%‡</td>
<td>51.59%</td>
</tr>
<tr>
<td></td>
<td>75.88%</td>
<td>73.15%†</td>
</tr>
</tbody>
</table>
Parameter Tuning

- Rank parameter $r$ and regularization parameter $\lambda$
- Rank parameter

![Graph showing Mean Average Precision (MAP) vs Rank (r) with lambda = 0.01](image)
Summary

• Incorporate type information in tensor decomposition

• Develop tricks to compute inversion to handle regularization efficiently

• Outperforms existing algorithm in knowledge base completion tasks
Thanks!
Supplementary

- Implementing TRESCAL along on relation extraction would not lead to good results

- (Not much improved if combined)
Supplementary

- Could be resulted from the problem setting
- RI13 (best) only consider entity pairs that have only occurred in the database