

# Typed Tensor Decomposition of Knowledge Bases For Relation Extration

**Kai-Wei Chang: University of Illinois**

**Bishan Yang: Cornell Univer.**

**Wen-tau Yih, Christopher Meek: Microsoft Research**

2014 年 12 月 4 日

# Outline

- RE introduction
- Related word(other approach)
- background
  - Tensor encoding
  - RESCAL
- Approach(improvement)
  - Domain Knowledge
  - Regulation efficiency
- Experiment
  - Knowledge base Completion
  - Relation extraction
- Conclusions

# RE introduction

- Tradition

- 分类问题，利用文本特征
- 没有利用已有 KB 的知识帮助判断

- Later

- (collective filtering?) 把 relation 转化为 vector，实体对和 relation 建立关系，映射到相同的 vector 上，然后计算相似度

	$r_1$	$r_2$	$r_3$
(e1,e2)	1		
(e2,e4)		1	
(e1,e3)	1		

- 没有挖掘 entity 信息，只是用序号代替 entity

- TRESICAL

- 利用 relation 限制和 entity 类别来剪枝
- 运行速度快，不影响关系抽取效果

## Related word(other approach)

### ● 张量分解

- CP(可分布式) or Tucker decompositions
- triples->tensor;CP decomposotion;extract hidden triples
- 改进: 加入 entity 类别信息有关的约束到目标函数中 (基于无参的 3-way tensor 贝叶斯模型)
- 大多数 TD model 空间大, 慢不实用

### ● 神经网络

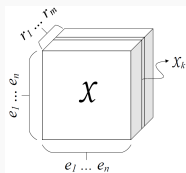
- 1. 学习  $(e_i, r_k)$  的向量表示
- 2. 对于任意  $(e_i, r_k, e_j)$ , 得到
$$\text{Vector}(e_i, r_k), \text{Vector}(e_j, r_k), \text{Score}(e_i, r_k, e_j) = v(e_i, r_k) \cdot v(e_j, r_k)$$
  - $\text{Score}(e_i, r_k, e_j) = -\|\mathbf{e}_i + \mathbf{r}_k - \mathbf{e}_j\|$

### ● 神经网络 2

- 1. 学习单词的向量 vector, 每个关系的矩阵  $R$
- 2.  $\text{Score}(e_i, r_k, e_j) = \text{vector}(e_i) \mathcal{R}_k \text{vector}^T(e_j)$

## background

- triples  $\rightarrow$  tensor



- $\mathcal{X}_k \approx \mathbf{A} \mathcal{R}_k \mathbf{A}^T$
- $[\mathbf{A}]_{n \times r}, [\mathcal{R}_k]_{r \times r}$

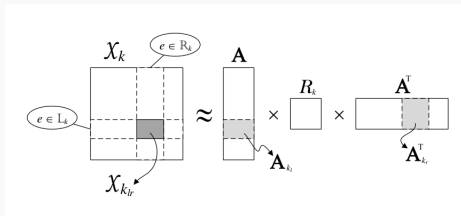
$$\min_{\mathbf{A}, \mathcal{R}_k} f(\mathbf{A}, \mathcal{R}_k) + \lambda \cdot g(\mathbf{A}, \mathcal{R}_k), \quad (2)$$

where  $f(\mathbf{A}, \mathcal{R}_k) = \frac{1}{2} (\sum_k \|\mathcal{X}_k - \mathbf{A} \mathcal{R}_k \mathbf{A}^T\|_F^2)$  is the mean-squared reconstruction error and  $g(\mathbf{A}, \mathcal{R}_k) = \frac{1}{2} (\|\mathbf{A}\|_F^2 + \sum_k \|\mathcal{R}_k\|_F^2)$  is the regularization term.

- 优化: 交替优化 (ALS)

## Approach (improvement)

- Domain Knowledge, 加入类别判断
  - (person, born-in, person) 剔除
  - $\mathcal{X}, \mathbf{A}$  的维数都减少



- SVD 分解 [p5 右中]
  - 原来的瓶颈: 求  $[(\mathbf{Z}^T \mathbf{Z} + \lambda \mathbf{I})^{-1}]_{\tau^2 \times \tau^2}$  [p4 左中]
  - 替代为求  $\mathbf{A}_{k_{lr}}$  的 SVD 分解
- 速度提升为 4 倍多

# Experiment

- KB completion

- NELL:v165(training), v166/533(development), v534/745(test)
- Entity retrieval:  $(e_i, r_k, ?)$ 
  - 1 个真实答案  $e_j$ +100 个随机挑选的实体  $e'_1, \dots, e'_{100}$ , 在其中查找答案
  - 2.  $Score(e_i, r_k, e_j) = (a^T)_i R_k a^j$

- Relation Retrieval:  $(e_i, ?, e_j)$

- $Score(e_i, r_k, e_j) = (a^T)_i R_k a^j$ , Domain 效果不好
  - 作者解释: entity 类别表错导致准确率降低 (直接被筛掉了)
- 含有相似 relation 的 relation 比较容易判断出来
  - 上位词, 抽象词的关系更容易表示?

	Entity Retrieval			Relation Retrieval		
	TransE	RESICAL	TRESCAL	TransE	RESICAL	TRESCAL
w/o type checking	51.41% <sup>‡</sup>	51.59%	54.79%	75.88%	73.15% <sup>‡</sup>	76.12%
w/ type checking	67.56%	62.91% <sup>‡</sup>	69.26%	70.71% <sup>‡</sup>	73.08% <sup>‡</sup>	75.70%

## Experiment and Conclusions

- Relation extraction

- 比如 RI13, 对于任意 relation  $r_k$ , 利用 RI13 返回的前 1000 个实体 ( $e_i, e_j$ ) 对作为候选, 取得分最高的前 100 个作为本系统的输出结果, 并比较正确率 (发现有明显提高)

Relation	#	MI09	YA11	SU12	RI13	TR	TR+SU12	TR+RI13
person/company	171	0.41	0.40	0.43	0.49	0.43	0.53	<b>0.64</b>
location/containedby	90	0.39	0.43	0.44	0.56	0.23	0.46	<b>0.58</b>
parent/child	47	0.05	0.10	0.25	0.31	0.19	0.24	<b>0.35</b>
person/place_of_birth	43	0.32	0.31	0.34	0.37	0.50	0.61	<b>0.66</b>
person/nationality	38	0.10	<b>0.30</b>	0.09	0.16	0.13	0.16	0.22
author/works_written	28	0.52	0.53	0.54	<b>0.71</b>	0.00	0.39	0.62
person/place_of_death	26	0.58	0.58	0.63	0.63	0.54	0.72	<b>0.89</b>
neighborhood/neighborhood_of	13	0.00	0.00	0.08	0.67	0.08	0.13	<b>0.73</b>
person/parents	8	0.21	0.24	<b>0.51</b>	0.34	0.01	0.16	0.38
company/founders	7	0.14	0.14	0.30	0.39	0.06	0.17	<b>0.44</b>
film/directed_by	4	0.06	0.15	0.25	0.30	0.03	0.13	<b>0.35</b>

- Conclusions

- 类别信息很有效, 准确率提高, 实验复杂度降低