Injecting Logical Background for Relation Extraction

$NAACL2015$

Tim Rockt aschel (UCL)
Sameer Singh (UW)
Sebastian Riedel (UCL)

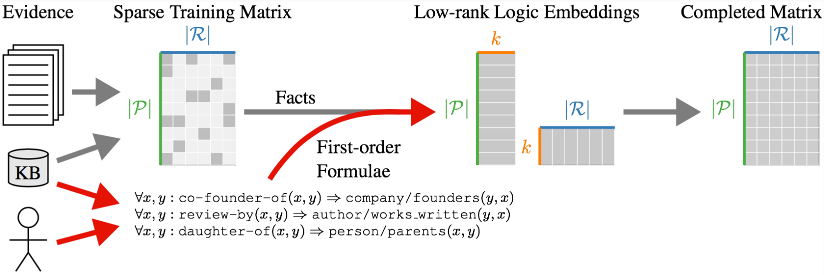
University College London

2015 年 5 月 8 日

Outline

- 论文概述
- 矩阵分解 (Matrix factorization), 一阶谓词逻辑 (first-order logic)
- 符号集 (Notation)
- 一阶谓词逻辑注入 (Inject) 矩阵分解
    - 矩阵分解之前操作 (Pre-Factorization)
    - 融入矩阵分解之中 ('joint')
- 实验
    - 预处理, 规则生成, 评价

summary



原始矩阵 $|\mathcal{P}| \times |\mathcal{R}|$

	fr1	fr2	fr3	p1	p2	p3	p4	p5	p6
(e1,e2)	1	0	0	0	1	0	0	1	0
(e2,e3)	0	1	0	1	0	0	1	0	0
(e1,e3)	0	0	1	0	0	1	0	0	0
(e1,e10)	1	0	0	1	0	0	0	0	0
(e1,e6)	1	1	0	1	0	0	1	1	0
(e1,e8)	1	0	0	1	0	0	0	0	0
(e3,e7)	0	1	0	1	0	0	1	0	0

- 矩阵分解 (MF) vs 规则抽取 (Rule-based)
  - MF: 开放域、不需要标注数据（弱监督）；完全依赖于 KB 的质量
    - 关系出现次数多，预测才好；不出现的无法预测；
  - Rule-based: 可以发现新的实体，新的关系；噪声大
- 本文动机
  - 矩阵分解和规则（一阶逻辑）混合
- 本文的贡献
  1. 如何在矩阵分解前、后利用谓词逻辑
  2. 利用一阶谓词逻辑训练更好的实体对向量、谓词向量
  3. 结合 MF，FOL 来预测实体的关系

Notation

$\mathcal{P} \subseteq \varepsilon \times \varepsilon$: 实体对 (e_i, e_j) 集合

$\mathcal{R}$: 关系 fr_i, p_j 集合

V: 向量空间 $\{v_{fr_1} = \mathbf{v_{r_m}}, v_{fr_2}, ..., v_{p_1} = \mathbf{v_{(e_i,e_j)}}, v_{p_2}, ...\}$

$\pi_m^{e_i,e_j} = \sigma(\mathbf{v_{r_m}}, \mathbf{v_{(e_i,e_j)}})$: 实体对与关系的相似性 σ:sigmoid func

- 一阶谓词
  - 原子问题（ground Atom）: $professorAt(x, y)$
  - 复合问题（logical background knowledge）:
    $professorAt(x, y) \Rightarrow emplyeeAt(x, y)$

评价一阶谓词子集 $\mathbf{w}$ 的好坏

$$p(\mathbf{w}|V) = \prod_{r_m(e_i,e_j) \in w} \pi_m^{e_i,e_j} \prod_{r_m(e_i,e_j) \notin w} (1 - \pi_m^{e_i,e_j})$$

Method

1. 传统方法
   - 直接矩阵分解（MF）
   - 只利用逻辑规则（Inf）
2. FOL,MF “混合”模型
   1. FOL 作为预处理（Pre-Factorization Inference）
      - 先用 FOL 改变原始矩阵 $M_{|\mathcal{P}|\times|\mathcal{R}|}$ ，再进行普通矩阵分解
   2. 将抽取的规则（一阶谓词子集）加入目标函数（Joint Optimization）
   3. 传统矩阵分解生成向量空间 V，然后使用收集的逻辑规则预测（Post-Factorization Inference）

一阶逻辑 (FOL) 注入

两种途径

1. FOL 作为预处理（Pre-Factorization Inference）
   - 先用 FOL 改变原始矩阵 $M_{|\mathcal{P}|\times|\mathcal{R}|}$，再进行普通矩阵分解
2. 一阶谓词子集加入目标函数（Joint Optimization）

Pre-Factorization Inference

(FOL 作为预处理)

Pre-Factorization Inference(FOL 作为预处理)

核心：弥补矩阵的稀疏性，增加收集的一阶逻辑规则 $\mathfrak{F}$与 freebase relation 之间的关联

方法：根据逻辑规则，增加原始矩阵 $M_{|\mathcal{P}|\times|\mathcal{R}|}$ 中值为 1 的元素个数

steps

1. 收集一阶谓词规则
2. 对每一条规则，比如 $\mathbf{F} = \forall x, y : r_s(x, y) \Rightarrow (x, y)$
   - 如果 $[(x, y), r_s]$ 上的值为 1，则把 $[(x, y), r_t]$ 上的值置为 1
   - 递归做下去，知道没有新的 1 出现
3. 对更新后的矩阵 $M'_{|\mathcal{P}|\times|\mathcal{R}|}$ 进行传统矩阵分解

Joint Optimization(组合模型)

Joint Optimization(组合模型)

核心：把收集的一阶逻辑规则 $\mathfrak{F}$加入到矩阵分解的目标函数中，训练出更好的属性向量 v_{r_m}、实体对向量 $v_{(e_i,e_j)}$

训练集的目标函数：

$$\min_V \sum_{\mathcal{F}\in\mathfrak{F}} \mathcal{L}([\mathcal{F}])$$

$[\mathcal{F}]$ is the marginal probability $p(w|V)$ that the formula F is true under the model

$\mathcal{L}([\mathcal{F}] := -\log([\mathcal{F}])$

V：向量空间 $\{v_{fr_1} = \mathbf{v_{r_m}}, v_{fr_2}, ..., v_{p_1} = \mathbf{v_{(e_i,e_j)}}, v_{p_2}, ...\}$

$p(\mathbf{w}|V) = \prod_{r_m(e_i,e_j)\in w} \pi_m^{e_i,e_j} \prod_{r_m(e_i,e_j)\notin w} (1 - \pi_m^{e_i,e_j})$

边缘分布 (marginal probability)

definition

对多变量的分布函数，针对某个变量进行求和(枚举其他变量的所有情况)，所得到的概率分布

example

联合概率 $P(A,B)$，对 A 的边缘分布为 $P_1(A) = \sum_{b\in B} P(A,b)$，对 B 的边缘分布为 $P_2(B) = \sum_{a\in A} P(a,B)$

边缘分布 (marginal probability)

$$p(\mathbf{w}|V) = \prod_{r_m(e_i,e_j)\in w} \pi_m^{e_i,e_j} \prod_{r_m(e_i,e_j)\notin w} (1-\pi_m^{e_i,e_j})$$
$$p(\mathbf{w}|V) = \prod f_{r_m(e_i,e_j)} = f_{r_m(e_i,e_j)} \times f_{r_n(e_k,e_l)} \times \ldots$$

$$f_{r_m(e_i,e_j)} = \begin{cases} \pi_m^{e_i,e_j} & \text{if } r_m(e_i,e_j) \in w \\ 1-\pi_m^{e_i,e_j} & \text{if } r_m(e_i,e_j) \notin w \end{cases} \tag{1}$$

- 当 $\mathcal{F}$ 是原子命题时 $[\mathcal{F}] = \pi_m^{e_i,e_j}$
- 当 $\mathcal{F} = \mathcal{A} \wedge \mathcal{B}$, $\mathcal{A}$ 和 $\mathcal{B}$ 是原子命题时, $[\mathcal{F}] = \pi_m^{e_i,e_j} \times \pi_n^{e_k,e_l} = [\mathcal{A}][\mathcal{B}]$
- 当 $\mathcal{F} = \neg\mathcal{A}$, $\mathcal{A}$ 是原子命题时, $[\mathcal{F}] = 1 - [\mathcal{A}]$

$\forall[\mathcal{F}]$

$$\mathcal{F} = \mathcal{A} \vee \mathcal{B} = 1 - (\neg\mathcal{A}) \wedge (\neg\mathcal{B})$$
$$[\mathcal{A} \vee \mathcal{B}] = [\mathcal{A}] + [\mathcal{B}] - [\mathcal{A}][\mathcal{B}]$$
$$[\mathcal{A} \Rightarrow \mathcal{B}] = [\mathcal{A}]([\mathcal{B}] - 1) + 1 \ldots$$

Joint Optimization(组合模型)

目标函数：

$$\min_V \sum_{\mathcal{F} \in \mathfrak{F}} \mathcal{L}([\mathcal{F}])$$

使用随机梯度下降（stochastic gradient descent）求解

- $\mathcal{L}([\mathcal{F}] \ := -\log([\mathcal{F}])$
- 对任意一个 $\mathcal{F}$, 其只和实体对向量、谓词向量有关
- 对任意 $\mathcal{L}([\mathcal{F}])$ 只要求 $\partial \mathcal{L}([\mathcal{F}])/\partial v_{r_m}$ 和 $\partial \mathcal{L}([\mathcal{F}])/\partial v_{(e_i,e_j)}$

Joint Optimization(组合模型)

$$\pi_m^{e_i,e_j} = \sigma(\mathbf{v}_{\mathbf{r_m}}, \mathbf{v}_{(\mathbf{e_i},\mathbf{e_j})})$$
$$\mathcal{L}([\mathcal{F}] := -\log([\mathcal{F}])$$

$\mathcal{F} = \pi_m^{e_i,e_j}$ is a ground atom

$$\partial[\mathcal{F}]/\partial v_{r_m} = [\mathcal{F}](1-[\mathcal{F}])v_{(e_i,e_j)}$$
$$\partial[\mathcal{F}]/\partial v_{(e_i,e_j)} = [\mathcal{F}](1-[\mathcal{F}])v_{r_m}$$
$$\partial\mathcal{L}([\mathcal{F}])/\partial v_{r_m} = -[\mathcal{F}]^{-1}\partial[\mathcal{F}]/\partial v_{r_m}$$
$$\partial\mathcal{L}([\mathcal{F}])/\partial v_{(e_i,e_j)} = -[\mathcal{F}]^{-1}\partial[\mathcal{F}]/\partial v_{(e_i,e_j)}$$

$\mathcal{F}$ is a First-order Logic

e.g.

$$\mathcal{F} = \forall x, y : r_s(x, y) \Rightarrow t(x, y)$$
$$[\mathcal{F}] = \prod_{(e_i,e_j)\in\mathcal{P}} [\mathcal{F}_{ij}]$$
$$\mathcal{L}([\mathcal{F}]) = \sum_{(e_i,e_j)\in\mathcal{P}} \mathcal{L}([\mathcal{F}_{ij}])$$

目标函数：

$$\min_V \sum_{\mathcal{F}\in\mathfrak{F}} \mathcal{L}([\mathcal{F}])$$

Formulae Extraction and Annotation(一阶逻辑规则 $\mathfrak{F}$ 抽取)

1. 由分解原始矩阵 $M_{|\mathcal{P}|\times|\mathcal{R}|}$ 得到实体对向量、关系向量
2. 对所有关系对：$(r_s, r_t), r_t \in freebase$
   1. 对训练集中的所有实体对 (e_i, e_j)，计算 $[r_s(e_i, e_j) \Rightarrow r_t(e_i, e_j)]$
3. 取值最高的 100 个作为抽取的一阶逻辑规则集合 $\mathfrak{F}$
   - 其中前 36 个比较好

Formula

0.97 $\forall x, y : \#2 - unit - of - \#1(x, y) \Rightarrow org/parent/child(x, y)$

0.97 $\forall x, y : \#2 - city - of - \#1(x, y) \Rightarrow location/containedby(x, y)$

0.97 $\forall x, y : \#2 - minister - \#1(x, y) \Rightarrow person/nationality(x, y)$

0.96 $\forall x, y : \#2 - minister - \#1(x, y) \Rightarrow person/nationality(x, y)$

Experiment

Experiment

1. Zero-shot Relation Learning

- 舍去所有 freebase relation 信息,训练数据中 *freebase relation* 的列都置为 *0*(模拟新关系的发现)

2. Relations with Few Distant Labels

- 给出部分 freebase relation 信息.
  给出率为 0 时退化为实验 1

3. Comparison on Complete Data

- 给出全部 freebase relation 信息.
  等价于实验 2 中给出率为 100%

原始矩阵 $|\mathcal{P}| \times |\mathcal{R}|$

	fr1	fr2	fr3	p1	p2
(e1,e2)	1	0	0	0	1
(e2,e3)	0	1	0	1	0
(e1,e3)	0	0	1	0	0
(e1,e10)	1	0	0	1	0
(e1,e6)	1	1	0	1	0
(e1,e8)	1	0	0	1	0
(e3,e7)	0	1	0	1	0

Experiemnt

由训练集 (**原始的矩阵**) 提取规则集合 $\mathfrak{F}$

目标函数 $\min_V \sum_{\mathcal{F}\in\mathfrak{F}} \mathcal{L}([\mathcal{F}])$

利用**过滤后的矩阵**，分解得到初始实体对向量 $v_{(e_i,e_j)}$，关系向量 v_{r_m}

随机梯度下降解得实体对向量 $v_{(e_i,e_j)}$，关系向量 v_{r_m}

- 不同实验共享相同规则集 $\mathfrak{F}$
- 不同实验“过滤后的矩阵”不同
- 一般矩阵分解目标函数 $\min_V \sum_{((i,j)} |M(i,j)) - v_i \cdot v_j|^2$
- 得到最终的向量空间后，判断 (e_i, e_j) 是否含有 r_t 关系只要判断 $\sigma(\mathbf{v_{r_m}}, \mathbf{v_{(e_i,e_j)}})$ 是否大于阈值

weighted Mean Avarage Precision(wMAP)

MAP

= relevant documents for query 1

Ranking #1

Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

= relevant documents for query 2

Ranking #2

Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

$$\textit{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$

$$\textit{average precision query 2} = (0.5 + 0.4 + 0.43)/3 = 0.44$$

$$\textit{mean average precision} = (0.62 + 0.44)/2 = 0.53$$