

Single-Step Inference Based on Distributed Representation

Qianni Wan, Jianliang Xu

College of Information Science and Engineering, Ocean University of China, Qingdao Shandong
Email: 1526162857@qq.com

Received: Mar. 5th, 2020; accepted: Mar. 20th, 2020; published: Mar. 27th, 2020

Abstract

At present, most of the existing knowledge graphs have the problem of lack of information. As one of the key technologies of knowledge graph, relational inference can complete the in-depth analysis and mining of data relations, which is a hot issue in the field of knowledge graph. This paper mainly introduces several common single step inference algorithms based on distributed representation, and describes the main process and evaluation index of the algorithm in order to provide reference for future research.

Keywords

Distributed Representation, Single Step Inference, Relation Inference

基于分布式表示的单步推理

万倩妮, 徐建良

中国海洋大学信息科学与工程学院, 山东 青岛
Email: 1526162857@qq.com

收稿日期: 2020年3月5日; 录用日期: 2020年3月20日; 发布日期: 2020年3月27日

摘要

目前, 现有知识图谱大多存在信息缺失的问题, 作为知识图谱关键技术之一的关系推理可以完成对数据关系的深入分析和挖掘, 是知识图谱领域的研究热点。本文主要介绍了几种常用的基于分布式表示的单步推理算法, 概述了算法的训练过程和评价指标, 为今后该领域的研究提供参考借鉴。

关键词

分布式表示, 单步推理, 关系推理



1. 引言

单步推理是通过知识图谱的知识要素(事实元组)来进行学习推理的,其特点是没有考虑路径特征且是直接关系。依据不同的推理方式,单步推理分成基于规则的单步推理、基于分布式表示的单步推理和基于神经网络单步推理。基于分布式的知识表示分为基于翻译、张量/矩阵分解和空间分布等多种方法[1]。

2. 分布式表示

基于分布式的知识表示是指将三元组中的实体和关系分别转换成稠密低维的实值向量。分布式的知识表示包括实体向量和关系向量两种向量,其中实体向量既可以表示头实体也可以表示尾实体。知识图谱中的两个实体和其之间的关系用 (h, r, t) 形式的三元组表示,黑斜体的 h, t, r 分别表示头实体向量、尾实体向量和关系向量。

3. 常见的基于分布式表示的单步推理算法

基于分布式表示的单步推理是指在分布式知识表示的基础上进行的直接推理过程。其中,最典型的是基于翻译的表示推理。该方法的思想来源于 word2vec 实验结果的启发。Mikolov 等人提出了 word2vec 词表示学习模型和工具包[2] [3],在其基础上发现了训练好的词向量之间存在平移不变的现象,并且通过类比推理实验发现,这种平移不变的现象普遍存在于词汇的语义关系之中[4]。

(1) TransE 模型

受到该现象的启发, Bordes 等人在 2013 年的 NIPS 大会上提出了第一个基于翻译的表示模型 TransE [5]。TransE 模型的主要思想是:如果三元组 (h, r, t) 成立,则头实体向量 h 与关系向量 r 的和与尾实体向量 t 接近(即 $h + r \approx t$)。否则远离。由上述基本假设得到势能函数:

$$f_r(h, t) = \|h + r - t\|_{L_n}$$

其中: L_n 表示 L_n 范数, n 的取值为 1 或者 2。TransE 通过定义最大间隔的方法,定义了损失函数:

$$L = \sum_{(h, r, t) \in S} \sum_{(h', r, t') \in S'_{(h, r, t)}} [\gamma + f_r(h, t) - f_r(h', t')]_+$$

其中, $[x]_+$ 表示 $\max(0, x)$; γ 表示正负样本之间的间隔,是一个常数;负样本集合 S' 表示为:

$$S'_{(h, r, t)} = \{(h', r, t) | h' \in E\} \cup \{(h, r, t') | t' \in E\}$$

与以往模型相比, TransE 模型的参数较少,计算复杂度低,性能有显著提升。但是 TransE 模型存在一些不足:对于不同的关系,所有实体的向量表示都相同;在解决知识图谱中一对一关系的数据时效果明显,而在处理复杂关系(一对多、多对一、多对多)时会遇到一定的困难。由此,掀起了基于分布式知识表示的研究热潮。

(2) TransH 模型

针对 TransE 模型处理复杂关系的弊端,2014 年 Wang 等人提出了 TransH 模型[6],引入了关系超平面,将关系解释为超平面上的翻译操作。TransH 模型的主要思想是:给定一个三元组 (h, r, t) ,对于每一个关系向量 r ,都能确定一个与其对应的超平面,然后将该三元组中的头实体向量 h 与尾实体向量 t 投影

到该超平面上, 将头实体向量在超平面上的投影记为 h_{\perp} , 将尾实体向量在超平面上的投影记为 t_{\perp} , 可以得到:

$$h_{\perp} = h - w_r^T h w_r, \quad t_{\perp} = t - w_r^T t w_r$$

其中: w_r 表示关系 r 所确定超平面的法向量, 为了简化计算过程, 使 $\|w_r\| = 1$ 。

如果三元组成立, 则 $h_{\perp} + r \approx t_{\perp}$ 。基于该假设进行建模, 得到 TransH 的势能函数:

$$f_r(h, t) = \|h_{\perp} + r - t_{\perp}\|_{L_n}$$

(3) TransR 和 CTransR 模型

TransE 和 TransH 模型都将实体和关系放到同一个语义空间中, 从而限制了模型的表达能力。针对这一问题, Lin 等人则提出了 TransR 模型[7], 在独立的实体空间和关系空间中分别建立实体向量和关系向量。首先将实体从实体空间投影到对应的关系空间, 然后在投影实体之间进行关系的翻译。TransR 模型的主要思想是: 给定一个三元组 (h, r, t) , 将头实体和尾实体向量投影到该关系空间得到:

$$h_r = M_r h, \quad t_r = M_r t$$

其中, $M_r \in R^{k \times d}$ 是实体空间到关系对应空间的投影矩阵; R^d 是实体空间; R^k 是关系空间。TransR 的势能函数为:

$$f_r(h, t) = \|h_r + r - t_r\|_{L_n}$$

TransE、TransH 和 TransR 模型为每个关系学习一个唯一的向量, 由于这些关系通常是不同的, 可能不足以表示该关系下的所有实体对。为了更好地表示关系, CTransR [7]模型采用分段线性回归方法拓展了 TransR 模型。CTransR 模型的主要思想是: 对于一个特定的关系 r , 训练数据中的所有实体对 (h, t) 被聚集成多个分组, 并且每个分组中的实体对都期望显示相似的关系 r 。所有的实体对 (h, t) 都用它们的向量偏移量 $(h - t)$ 表示以进行聚类, 并为每一个分组或者子关系 r_c 分别学习向量表示。将头实体和尾实体向量投影到该关系空间得到:

$$h_{r,c} = M_r h, \quad t_{r,c} = M_r t$$

其中: $M_r \in R^{k \times d}$ 是实体空间到关系对应空间的投影矩阵。CTransR 的势能函数为:

$$f_r(h, t) = \|h_{r,c} + r_c - t_{r,c}\|_{L_n} + a \|r_c - r\|_{L_n}$$

其中: $\|r_c - r\|_{L_n}$ 是为了保证 r_c 距离 r 不会太远; a 是控制约束效果的参数。

(4) TransD 模型

TransD 模型用两个向量来表示实体和关系, 一个用于表示实体的含义, 另一个用于动态构造映射矩阵[8]。与 TransR/CTransR 相比, TransD 模型不仅考虑了关系的多样性, 而且考虑了实体的多样性。TransD 参数少, 不需要矩阵向量乘法运算。TransD 模型的主要思想是: 对于给定的三元组 (h, r, t) , 它的向量分别为 h, h_p, r, r_p, t, t_p , 下标 p 表示投影向量, 定义两个投影矩阵将实体从实体空间映射到关系空间:

$$M_{rh} = r_p h_p^T + I^{m \times n}, \quad M_{rt} = r_p t_p^T + I^{m \times n},$$

其中: $h, h_p, t, t_p \in R^n$, $r, r_p \in R^m$, $M_{rh}, M_{rt} \in R^{m \times n}$ 。因此, 映射矩阵是由实体和关系共同决定的, 这种操作使得两个投影向量能够很好地相互作用, 因为它们的每个元素都能满足来自另一个向量的每个元素。当我们初始化每个映射矩阵时, 我们将 $I^{m \times n}$ 添加到 M_{rh} 和 M_{rt} 中。利用映射矩阵, 我们将投影向量定义如下:

$$h_{\perp} = M_{rh} h, \quad t_{\perp} = M_{rt} t$$

综上所述, 得到其势能函数为:

$$f_r(h, t) = \|h_{\perp} + r - t_{\perp}\|_{L_n}$$

(5) TransM 模型

与上述模型相比, TransM 模型更关注每个训练三元组对优化目标的不同贡献(即各种关系映射属性)[9]。TransM 模型定义了与关系相关的权重, 将训练的每个三元组与权重相关联。对于给定的三元组 (h, r, t) , TransM 的势能函数为:

$$f_r(h, t) = w_r \|h + r - t\|_{L_n}$$

(6) TranSparse 模型

上述的模型都没有考虑到知识图谱翻译的异质性和不平衡性, 因此 Guoliang Ji 等人提出来 TranSparse 模型[10], 使用自适应稀疏矩阵来建模不同类型的关系。在 TranSparse 模型中, 翻译矩阵被自适应稀疏矩阵所代替, 其稀疏度由关系连接的实体(或实体对)的个数决定, 且关系的两边共享相同的翻译矩阵。TranSparse 模型包括 TranSparse (share)和 TranSparse (separate)两种模型。

TranSparse (share)模型的主要思想是: 为每个关系 r 设置一个稀疏矩阵 $M_r(\theta_r)$, $M_r(\theta_r)$ 的稀疏度 θ_r 定义为:

$$\theta_r = 1 - (1 - \theta_{\min}) N_r / N_r^*$$

其中: N_r 表示关系 r 连接的实体对的数量; N_r^* 表示连接实体对的最大数量(即 N_r 的最大值); $0 \leq \theta_{\min} \leq 1$ 。

头实体和尾实体共享一个稀疏矩阵, 头实体和尾实体的投影向量可以定义为:

$$h_p = M_r(\theta_r)h, t_p = M_r(\theta_r)t$$

其中: $M_r(\theta_r) \in R^{m \times n}$; $h, t \in R^n$; $h_p, t_p \in R^m$ 。

TranSparse (separate)模型中, 为每个关系设置两个单独的稀疏矩阵 $M_r^h(\theta_r^h)$ 和 $M_r^t(\theta_r^t)$, $M_r^h(\theta_r^h)$ 和 $M_r^t(\theta_r^t)$ 的稀疏度定义为:

$$\theta_r^l = 1 - \frac{(1 - \theta_{\min}) N_r^l}{N_r^{l*}} (l = h, t)$$

其中: 下标 r 是关系的索引; h, t 是指矩阵用于哪个实体(头实体或尾实体); $N_r^l(l = h, t)$ 表示关系 r 在位置 l 处连接的实体数; N_r^{l*} 表示它们的最大数量(关系 r^* 在位置 l^* 连接的最多实体); $0 \leq \theta_{\min} \leq 1$ 。

头实体和尾实体的投影向量可以定义为:

$$h_p = M_r^h(\theta_r^h)h, t_p = M_r^t(\theta_r^t)t$$

TranSparse(share)和 TranSparse(separate)模型的势能函数为:

$$f_r(h, t) = \|h_p + r - t_p\|_{L_n}$$

其中: $r \in R^m$ 。

(7) TEKE 模型

Zhigang Wang 等人提出了 TEKE 模型[11], 引入了丰富的文本上下文信息。每个关系对于不同的头实体和尾实体拥有不同的表示, 以便更好地处理复杂关系。TEKE 模型包含四个关键部分: 实体注释、文本上下文翻译、实体/关系表示建模和训练。

在实体注释中, 对于给定的文本语料库 $\mathcal{D} = \langle w_1 \cdots w_i \cdots w_m \rangle$, 使用实体链接工具自动的标记实体, 然

后得到实体注释文本集 $\mathcal{D}' \langle x_1 \cdots x_i \cdots x_m \rangle$ 。其中, \mathcal{D}' 的长度 m' 小于 \mathcal{D} 的长度 m , 这是因为多个相邻的词可以标记为一个实体。

在文本上下文嵌入中, 为了将知识和文本信息联系在一起, 构建了一个基于实体注释文本集 \mathcal{D}' 的共现网络 $\mathcal{G} = (\mathcal{X}, \mathcal{Y})$ 。 $x_i \in \mathcal{X}$ 表示知识图谱中的节点/实体, 对应一个单词或者实体。 $y_{ij} \in \mathcal{Y}$ 表示 x_i 和 x_j 的共现频率。构建的共现网络 \mathcal{G} 能够利用丰富的文本信息将实体和词连接起来用于表示学习。将给定节点/实体 x_i 的上下文作为它的邻居, 定义为:

$$n(x_i) = \{x_j \mid y_{ij} > \theta\}$$

其中: θ 是一个阈值。

由于两个实体的共同邻居可以表示它们之间的关系, 因此将关系的文本上下文为两个实体的共同邻居, 定义为:

$$n(x_i, x_j) = \{x_k \mid x_k \in n(x_i) \cap n(x_j)\}$$

可以得到 \mathcal{G} 中的每个实体 x 的表示 $x \in R^k$, 基于实体向量表示, 定义了 x_i 的实体文本上下文翻译作为 $n(x_i)$ 中实体向量的加权平均值:

$$n(x_i) = \frac{1}{\sum_{x_j \in n(x_i)} y_{ij}} \sum_{x_j \in n(x_i)} y_{ij} x_j$$

同样地, 定义了 x_i 和 x_j 的关系文本上下文翻译作为 $n(x_i, x_j)$ 中关系向量的加权平均值:

$$n(x_i, x_j) = \frac{1}{Z} \sum_{x_k \in n(x_i, x_j)} \min(y_{ik}, y_{jk}) x_k$$

其中: 每个共同邻居的权重 $x_k = \min(y_{ik}, y_{jk})$; $Z = \sum_{x_k \in n(x_i, x_j)} \min(y_{ik}, y_{jk})$ 。

在文本上下文翻译中, 定义了头实体和尾实体的实体文本翻译的线性变换:

$$\hat{h} = n(h)A + h, \quad \hat{t} = n(t)A + t$$

其中: A 是一个 $k \times k$ 的矩阵, 可视为上下文的权重; h 和 t 分别为头实体向量和尾实体向量。

同样地, 定义了关系的关系文本翻译的线性变换:

$$\hat{r} = n(h, t)B + r$$

其中: B 是一个 $k \times k$ 的权重矩阵; r 为关系向量。

综上, 其势能函数定义为:

$$f_r(h, r, t) = \|\hat{h} + \hat{r} - \hat{t}\|_{L_n}$$

(8) TransA 模型

Yantao Jia 等人提出了 TransA 模型[12], 可以根据知识图谱的结构自适应地找到最佳损失函数, 并且不需要事先封闭的候选集。TransA 模型的主要思想是: 给定一个三元组 (h, r, t) , 势能函数定义为:

$$f_r(h, t) = \|h + r - t\|_{L_n}$$

其中: $(h, r, t) \in R^d$; d 表示向量空间的维数。

TransA 模型的重点在于自适应地找到最佳损失函数。由于关系集合和实体集合是两个不相交的集合, 因此定义了最佳间距 M_{opt} :

$$M_{opt} = \mu M_{ent} + (1 - \mu) M_{rel}$$

其中: $0 \leq \mu \leq 1$;

由上述公式可知, M_{opt} 分为实体间距和关系间距两部分, 并利用线性插值进行组合。实体最优间距 M_{ent} 应保证内层圆包含更多的正例, 外层圆的外部应尽可能是负例。 M_{ent} 的定义为: 给定的一个实体 h , 对于所有的 $t \in P_r$ 和 $t' \in N_r$, 则:

$$M_{ent} = \frac{\sum_{r \in R_h} \min_{t, t'} \sigma(\|h - t'\| - \|h - t\|)}{nr_h}$$

$$\sigma(x) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

其中: $P_r = \{t | (h, r, t) \in S\}$; $N_r = \{t | (h, r, t) \notin S, (h, r', t) \in S, \exists r' \in R\}$; S 是正确三元组的集合; nr_h 是实体 h 的关系数量; R_h 表示与 h 相关联的关系集合。

实体最优间距 M_{rel} 定义为: 对于给定一个实体 h 和与该实体相关联的一个关系 r , 如果存在 $r_i \in R_{h,r}$, 满足 $\|r_i\| \geq \|r\|$, 则:

$$M_{rel} = \min_{r_i \in R_{h,r}} (\|r_i\| - \|r\|)$$

其中: $R_{h,r} = \{r_1, r_2, \dots, r_{nr_{h,r}-1}\}$, 表示与 h 相关联的除关系 r 之外的关系集合。

由此, 损失函数定义为:

$$\mathcal{L}(\mathcal{A}_S, z_k) = \sum_{(h,r,t) \in \Delta} \sum_{(h',r',t') \in \Delta'} [f_r(h,t) + M_{opt} - f_r(h',t')]_+$$

其中: Δ 是正例; Δ' 是负例;

(9) TKRL 模型

鉴于大部分知识表示学习方法忽略了实体中丰富的层次类型信息, Ruobing Xie 等人认为不同类型的实体应该具有不同的表示, 由此提出了 TKRL 模型[13]。TKRL 模型为每个类型的实体设置由层次结构构造的特定类型投影矩阵 M , 然后在特定类型的投影下表示 h 和 t 。TKRL 模型的势能函数定义为:

$$f_r(h, r, t) = \|M_{rh}h + r - M_{rt}t\|_{L_n}$$

其中: M_{rh} 和 M_{rt} 是 h 和 t 不同的投影矩阵。

为了将层次类型信息编码到表示学习中, 提出了一种通用的类型编码器来构造每个实体的投影矩阵。其次, 提出了两种先进的编码器对层次结构中的内部连接和关系特定类型信息中的先验知识加以利用。其中, 实体 e 的投影矩阵 M_e 是所有类型矩阵的加权和。通用类型编码的投影矩阵定义为:

$$M_e = \alpha_1 M_{c_1} + \alpha_2 M_{c_2} + \dots + \alpha_n M_{c_n}$$

其中: n 表示实体 e 的类型数; c_i 是实体所属的第 i 个类型; M_{c_i} 是 c_i 的投影矩阵; α_i 表示 c_i 的相应权重。

对于通用类型的编码而言, 实体 e 的所有投影矩阵在不同情况下都是相同的。但是, 实体应该有不同的表示, 以强调在不同场景中更重要的属性。为了解决这一问题, 头实体的投影矩阵 M_{rh} 定义为:

$$M_{rh} = \frac{\sum_{i=1}^n \alpha_i M_{c_i}}{\sum_{i=1}^n \alpha_i}, \alpha_i = \begin{cases} 1, & c_i \in C_{rh} \\ 0, & c_i \notin C_{rh} \end{cases}$$

其中: C_{rh} 表示由关系特定类型信息给出的关系 r 中头实体的类型集; 同一个实体作为尾实体的投影矩阵 M_{rt} 与作为头实体时的 M_{rh} 相同; M_c 表示类型 c 的投影矩阵。

M_c 能够有以下两种方式编码:

a) RHE (Recursive Hierarchy Encoder)

$$M_c = \prod_{i=1}^m M_{c^{(i)}} = M_{c^{(1)}} M_{c^{(2)}} \cdots M_{c^{(m)}}$$

其中: m 是类型 c 在层次结构中的层数; $M_{c^{(i)}}$ 表示第 i 个子类型 $c^{(i)}$ 的投影矩阵。

b) WHE (Weighted Hierarchy Encoder)

$$M_c = \sum_{i=1}^m \beta_i M_{c^{(i)}} = \beta_1 M_{c^{(1)}} + \beta_2 M_{c^{(2)}} + \cdots + \beta_m M_{c^{(m)}}$$

其中: m 是层次结构中的层数; $M_{c^{(i)}}$ 是 $c^{(i)}$ 的投影矩阵; β_i 是 $c^{(i)}$ 的相应权重。

在 $c^{(i)}$ 和 $c^{(i+1)}$ 之间定义一个按比例下降的加权策略:

$$\beta_i : \beta_{i+1} = (1-\eta) : \eta, \quad \sum_{i=1}^m \beta_i = 1$$

其中: $\eta \in (0, 0.5)$ 。

与其它模型不同, TKRL 模型的损失函数定义为:

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'_{(h,r,t)}} [\gamma + f_r(h,r,t) - f_r(h',r',t')]_+$$

其中:

$$S'_{(h,r,t)} = \{(h',r,t) | h' \in E\} \cup \{(h,r,t') | t' \in E\} \cup \{(h,r',t) | r' \in R\}$$

(10) TAE 模型

由于现有的大多数方法都会忽略时间信息, 而只能从时间未知的事实三元组中学习。Tingsong Jiang 等人提出了 TAE 模型[14], 使用事实之间的时间顺序信息进行关系推理。TAE 模型使用 (e_i, r, e_j, t) 的四元组形式, 表示实体 e_i 和 e_j 在关系间隔 $t = [t_b, t_e]$ 具有关系 r 的事实 ($t_b \leq t_e$)。对于一些事实, 没有结束日期, 可以表示为 $t = [t_b, +\infty]$ 。

为了捕获关系的时间顺序, TAE 模型定义了一个时间演化矩阵 $T \in R^{n \times n}$ 来建模关系的演变。其中, n 是关系向量的维度; T 是模型学习数据的参数。TAE 模型的主要思想是: 给定一个 $(e_i, r_k, e_j, t_k) \in \Delta_i$, 可以找一个与时间相关的 $(e_i, r_l, e_m, t_l) \in \Delta_i$, 两个四元组共享同一个头实体和时序关系对 $\langle r_k, r_l \rangle$ 。如果 $t_k < t_l$, 定义一个正确的时序关系对 $y^+ = \langle r_k, r_l \rangle$ 和对应的错误的时序关系对 $y^- = \langle r_k, r_l \rangle^{-1} = \langle r_l, r_k \rangle$ 。因此, 时间势能函数定义为:

$$g(\langle r_k, r_l \rangle) = \|r_k T - r_l\|_{L_n}$$

注意: T 是非对称的。

对于给定的三元组 (e_i, r_k, e_j) , 其势能函数定义为:

$$f_r(e_i, r_k, e_j) = \|e_i + r_k - e_j\|_{L_n}$$

优化目标是优化联合势能函数, 损失函数定义为:

$$L = \sum_{x^+ \in \Delta} \left[\sum_{x^- \in \Delta'} [\gamma_1 + f_r(x^+) - f_r(x^-)]_+ + \lambda \sum_{y^+ \in \Omega_{e_i, r_k}, y^- \in \Omega'_{e_i, r_k}} [\gamma_2 + g(y^+) - g(y^-)]_+ \right]$$

其中: $x^+ = (e_i, r_k, e_j) \in \Delta$, 表示正例; $x^- = (e_i, r_k, e'_j) \in \Delta'$, 表示负例; Ω_{e_i, r_k} 表示为 (e_i, r_k, e_j, t_k) 的

正确的时序关系对的集合, 其定义为:

$$\Omega_{e_i, r_k} = \left\{ \langle r_k, r_l \rangle \mid (e_i, r_k, e_j, t_{r_k}) \in \Delta_t, (e_i, r_l, e_m, t_{r_l}) \in \Delta_t, t_{r_k} < t_{r_l} \right\} \\ \cup \left\{ \langle r_l, r_k \rangle \mid (e_i, r_k, e_j, t_{r_k}) \in \Delta_t, (e_i, r_l, e_m, t_{r_l}) \in \Delta_t, t_{r_k} > t_{r_l} \right\}$$

Ω'_{e_i, r_k} 是对 Ω_{e_i, r_k} 中每一个关系对求逆得到的。

(11) puTransE 模型

Yi Tay 等人提出了 puTransE 模型, 通过语义和结构感知选择机制生成多个表示空间[15], 每个表示空间在计数和多样性上都要满足事实三元组约束。对于每一个表示空间 $\Delta_i \in \Delta$, β_i 是表示空间 Δ_i 中的三元组约束(β_i 通常比三元组的总数 n_i 小), E_i 和 R_i 分别是表示空间 Δ_i 中实体集合和关系集合。

puTransE 模型重点在于三元组选择机制, 该机制包括主要包括语义感知的三元组选择、结构感知的三元组选择、语义和结构的平衡以及生成随机配置四个方面。

语义感知三元组的选择: 随机选取一个关系 r ($r \in R$), 生成所有包含关系 r 的所有实体集合 E_r^f , r 被视为 Δ_i 的语义焦点。

结构感知三元组的选择: 采用双向随机游走(RW)模型。

语义和结构的平衡: 由于 $E_r^f > \beta_i$ 的情况可能发生, 因此, 设置参数 $\theta \in [0, 1]$ 来调节语义和结构的平衡。

生成随机配置: 为每一个表示空间随机分配不同的参数。对于每一个 Δ_i , 随机生成 margin、学习率、 θ 、 β_i 和迭代批次。

puTransE 模型的全局势能函数定义为:

$$G_r(h, t) = \max_{(h, r, t) \in \Phi} -\|h + r - t\|_{L_n}$$

其中: Φ 是包含所有实体和关系的表示向量空间的集合;

损失函数定义为:

$$L = \sum_{\Delta_i \in \Delta} \sum_{\xi \in \Delta_i} \sum_{\xi' \notin \Delta_i} [E_i(\xi) + \gamma - E_i(\xi')]_+$$

其中: ξ 是存在于平行向量空间 Δ_i 的三元组; ξ' 是不存在于平行向量空间 Δ_i 的三元组。 $E_i(\xi)$ 是三元组在其向量空间的局部势能函数。

4. 算法的训练过程

算法的重点都在于训练的过程, 即采用向量表示的方法对三元组进行优化的过程, 因此接下来主要介绍基于分布式的单步推理算法的训练过程。训练流程图如图 1 所示:

1) 首先要初始化算法中使用的参数, 一般包括学习率、向量维度和损失函数中的间距参数等。

2) 初始化参数完成后, 读取训练集, 以便进行负例的构造。为了使正例与负例之间形成明显的对照关系, 因此在构造负例时, 可采用均匀采样法或伯努利采样法等随机替换正例中的头实体或尾实体或者关系。对于不同的推理算法, 由于替换的对象可能存在差异, 例如: 有些算法只随机替换头实体或尾实体, 而有些算法需随机替换头实体或尾实体或者关系, 会导致采样方法存在差异。

3) 构造完负例后, 根据不同算法的势能函数, 采用 L_n 范数计算正例与负例的势能函数。

4) 训练过程采用梯度下降算法加速训练过程, 利用正例和负例的势能函数值情况判断是否需要两组元组进行优化。优化时要尽量减小正例的势能值, 并尽量增大负例的势能值, 从而达到最小化损失函数的目的。

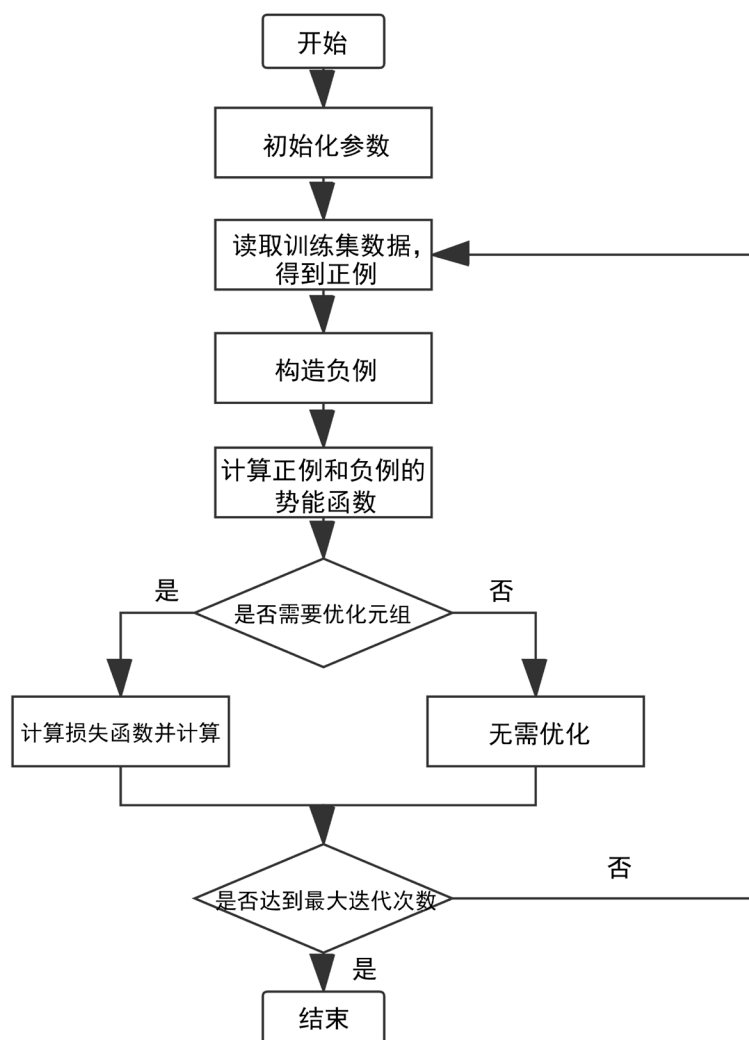


Figure 1. Training flowing-chart

图 1. 训练流程图

梯度下降法是一个最优化算法,常用于机器学习和人工智能当中用来递归性地逼近最小偏差模型[16]。梯度下降算法包括批量梯度下降(Batch gradient descent)、随机梯度下降(Stochastic Gradient Descent)和小批量梯度下降(Mini-batch gradient descent)三种[17]。批量随机梯度下降是指每次迭代使用所有样本进行梯度的更新,但当样本数据规模很大时,由于每次迭代需要计算所有的样本,因此极大地降低了模型的训练速度;随机梯度下降每次迭代只使用一个样本对参数进行更新,虽然可以加快模型的优化速度,但是由于单个样本无法代表样本整体,降低了结果的准确性;而小批量随机梯度下降综合了上述两种算法,每次迭代使用“batch_size”个样本对参数进行更新,不仅可以加快模型的优化速度,而且可以提高收敛结果的准确性。可以根据算法模型和数据集选择最合适的梯度下降算法。

5. 算法的评价标准

在算法评价指标方面,采用前 k 命中率(Hits@k)和正确实体的平均排名数(Mean Rank)两种指标分别进行实体预测和关系预测,以衡量算法的综合能力。

实体预测评价过程如下:假设知识图谱中有 n 个实体

(1) 将测试集中一个正确的三元组 a 的头实体或者尾实体, 依次替换为实体集中的所有实体, 即会产生 n 个三元组。

(2) 分别计算 n 个三元组的势能函数值, 这样可以得到 n 个势能函数值, 分别对应上述 n 个三元组。

(3) 对上述 n 个势能函数值进行升序排序, 记录三元组 a 的势能函数值排序后的序号 k 。

(4) 如果前 $k-1$ 个势能函数值对应的三元组中有 m 个三元组是正确的, 那么三元组 a 的序号改为 $k-m$ 。

(5) 对所有的正确的三元组重复上述过程。

上述过程结束后, 计算出排序后所有正确三元组序号的平均值, 得到的值称为 Mean Rank, 计算出排序后正确三元组的序号小于 k 的比例, 得到的值称为 Hits@k。

同理, 关系预测只是将测试集中正确三元组的关系, 依次替换为关系集中的所有关系, 再依次执行上述过程。

在算法的评测过程中, 可以根据实际需求, 选择评价指标对算法进行评估。不难发现, Mean Rank 越小, 证明该算法预测结果的准确性越高, Hits@k 越大, 证明该算法的预测结果越好。

6. 总结

基于分布式表示的关系推理技术能够很好地把握数据的内在特征, 算法高效, 易于操作, 解决了数据稀疏等问题。但是, 该类算法依赖于纯粹的数学驱动, 对实体之间的关系预测难以提供合理的解释。目前, 基于分布式表示的关系推理技术已经被广泛应用于机器翻译和智能问答等领域, 并且在金融反欺诈和数据异常分析等领域具有良好的应用前景。在大数据时代迅猛发展的今天, 丰富和拓展知识图谱是时代发展的必然趋势。如何对推理模型进行优化处理, 考虑知识图谱中更深层次的结构和语义信息, 提高推理算法的速度和准确率, 是未来该领域的主要研究方向。

参考文献

- [1] 官赛萍, 靳小龙, 贾岩涛, 王元卓, 程学旗. 面向知识图谱的知识推理研究进展[J]. 软件学报, 2018, 29(10): 2966-2994.
- [2] Mikolov, T., Sutskever, I., Chen, K., et al. (2013) Distributed Representations of Words and Phrases and Their Compositionality. In: *Proceedings of NIPS*, MIT Press, Cambridge, MA, 3111-3119.
- [3] Mikolov, T., Chen, K., Corrado, G., et al. (2013) Efficient Estimation of Word Representations in Vector Space. *Proceedings of ICLR*, arXiv: 1301.3781.
- [4] 刘知远, 孙茂松, 林衍凯, 谢若冰. 知识表示学习研究进展[J]. 计算机研究与发展, 2016, 53(2): 247-261.
- [5] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. and Yakhnenko, O. (2013) Translating Embeddings for Modeling Multi-Relational Data. In: *Proceedings of the Advances in Neural Information Processing System*, Curran Associates Inc., Red Hook, NY, 2787-2795
- [6] Wang, Z., Zhang, J., Feng, J., et al. (2014) Knowledge Graph Embedding by Translating on Hyperplanes. In: *Proceedings of AAAI*, AAAI, Menlo Park, CA, 1112-1119.
- [7] Lin, Y.K., et al. (2015) Learning Entity and Relation Embeddings for Knowledge Graph Completion. In: *Proceedings of AAAI*, AAAI, Menlo Park, CA, 2181-2187.
- [8] Ji, G., He, S., Xu, L., et al. (2015) Knowledge Graph Embedding via Dynamic Mapping Matrix. In: *Proceedings of ACL*, Stroudsburg, PA, 687-696. <https://doi.org/10.3115/v1/P15-1067>
- [9] Fan, M., Zhou, Q., Chang, E. and Zheng, T.F. (2014) Transition-Based Knowledge Graph Embedding with Relational Mapping Properties. In: *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation*, ACL, Stroudsburg, 328-337.
- [10] Ji, G., Liu, K., He, S. and Zhao, J. (2016) Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, AAAI, Menlo Park, CA, 985-991.
- [11] Wang, Z. and Li, J. (2016) Text-Enhanced Representation Learning for Knowledge Graph. In: *Proceedings of the 25th*

International Joint Conference on Artificial Intelligence, AAAI, Menlo Park, CA, 1293-1299.

- [12] Jia, Y., Wang, Y., Lin, H., Jin, X. and Cheng, X. (2016) Locally Adaptive Translation for Knowledge Graph Embedding. In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, AAAI, Menlo Park, CA, 992-998.
- [13] Xie, R., Liu, Z. and Sun, M. (2016) Representation Learning of Knowledge Graphs with Hierarchical Types. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, AAAI, Menlo Park, CA, 2965-2971.
- [14] Jiang, T., Liu, T., Ge, T., Sha, L., Chang, B., Li, S. and Sui, Z. (2016) Towards Time-Aware Knowledge Graph Completion. In: *Proceedings of the 26th International Conference on Computational Linguistics*, ACL, Stroudsburg, 1715-1724.
- [15] Tay, Y., Luu, A.T. and Hui, S.C. (2017) Non-Parametric Estimation of Multiple Embeddings for Link Prediction on Dynamic Knowledge Graphs. In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI, Menlo Park, CA, 1243-1249.
- [16] 人工智能-梯度下降法(1). <https://cloud.tencent.com/developer/news/238176>, 2018.
- [17] Ruder, S. (2016) An Overview of Gradient Descent Optimization Algorithms.