

TransX 表示 TransE 的多种变体, 如 TransH (Wang et al., 2014)、TransR (Lin et al., 2015b)和 STransE (Nguyen et al., 2016), 其中  $gr_i()$  表示关于关系  $r$  的矩阵乘法。

我们可以看到, TransE 能够推断和建模除对称模式之外的所有其他关系模式。原因是, 在 TransE 中, 任何对称关系都将由一个 0 平移向量表示。因此, 这将推动具有对称关系的实体在嵌入空间中彼此靠近。

此外, 由于抽样过程可能是昂贵的, 我们把上述概率作为负样本的权重。

KGGAN: 提出了一种生成式逆向学习框架来抽取负样本。然而, 这样的框架需要同时训练嵌入模型和离散负样本生成器, 这不仅难以优化, 而且计算开销也很大。

绘制关系嵌入中每个元素相位的直方图

Basic MF 是最基础的分解方式, 将评分矩阵  $R$  分解为用户矩阵  $U$  和项目矩阵  $S$ , 通过不断的迭代训练使得  $U$  和  $S$  的乘积越来越接近真实矩阵

预测值接近真实值就是使其差最小, 这是我们的目标函数, 然后采用梯度下降的方式迭代计算  $U$  和  $S$ , 它们收敛时就是分解出来的矩阵。

CP 分解是做了什么工作, 目的是想干什么。

CP 分解是将一个高维的张量, 分解成多个核的和, 每个核是由向量的外积组成; 通过这样的分解, 我们可以大大地降低参数的维度。

其实, 不止 CP 分解, 其他的张量分解算法都是同个道理, 只是所采用的分解方法不同而已。当然, 这样的分解只是原来张量的近似, 没办法保证完全复原。

从以上角度来说, 张量分解的目的跟矩阵分解是相同的, 只是一个是在二维上的分解, 另一个是在高维上的分解而已。

我们还可以将张量  $X$  矩阵化; 我们可以使用向量  $\lambda$  对三维向量  $a, b, c$  进行归一化:

当然, 这是当核是三维的情况下, 当核是  $N$  维时, 同个道理:

Tucker 分解 (Tucker decomposition) 是高阶的主成分分析的一种形式。它将一个张量分解成一个核张量与每一维矩阵的乘积, 具体如下:

$ABC$  是因子矩阵 (通常是正交的), 可以当做是每一维上的主要成分。核张量表示每一维成分之间的联系

实际上, 当核张量是对角化的且  $P=Q=R$  时, CP 分解便是 Tucker 分解的一种特殊形式。

同样的,  $N$  维张量可以表示成以下形式:

至于 Tucker 分解的优化方法, 有 HOSVD 和 HOOI 等方法

HOSVD: 利用 SVD 对每个 mode 做一次 Tucker1 分解 (截断或者不截断)。它通过张量的每一个 mode 上做 SVD 分解对各个 mode 上的因子矩阵进行求解, 最后计算张量在各个 mode 上的投影之后的张量作为核张量。HOSVD 不能保证得到一个较好的近似, 但 HOSVD 的结果

可以作为一个其他迭代算法 (如 HOOI) 的很好的初始解。将张量分解看作是一个优化的过程, 不断迭代得到分解结果。

RESCAL: 基于张量/矩阵分解的表示推理将 (头实体, 关系, 尾实体) 三元组看成张量/矩阵中的元素构建张量/矩阵, 通过张量/矩阵分解方法进行表示学习。分解得到的向量表示相乘重构成张量/矩阵, 元素值即为对应三元组有效与否的得分, 可以认为得分大于特定阈值的三元组有效, 或候选预测按照得分排序, 选择得分高的候选作为推理结果。

该类方法的典型代表是 Nickel 等人 [105] 提出的 RESCAL, 基于三阶张量进行表示学习, 模型如图 所示。如果三元组成立, 三阶张量上对应的元素值为 1, 否则为 0。例如, 图 2 中第  $i$  个实体和第  $j$  个实体存在第  $k$  种关系, 对应的位置  $(i, j, k)$  元素值为 1。该三阶张量对每种关系进行切片分解, 其中, 对第  $k$  种关系的切片, 如图 2 所示, 可以分解为实体表示矩阵  $A$ 、第  $k$  种关系对应的非对称矩阵  $R_k$  与实体表示矩阵的转置  $A^T$  的连乘。通过最小化重构误差学习实体和关系的表示。张量分解模型虽然推理准确率高, 但内存占用量大, 计算速度慢。

RESCAL 模型虽然具有很强的表达能力和强大的功能, 但由于参数较多, 容易过度拟合, 其嵌入维数随知识图中关系的个数呈二次增长

DistMult: 每一个关系都有对角矩阵的特殊情况。然而, 在 DistMult 中对主体实体嵌入向量所进行的线性变换仅限于拉伸。给定同一实体的主、客体实体嵌入的等价性, DistMult 学习到的三阶二元张量在主、客体实体模式下是对称的, 因此 DistMult 不能对非对称关系建模。

Simple: 张量分解方法被证明是一种有效的用于知识图谱补全的统计关系学习方法。最早提出的张量分解方法为 CP 分解 (canonical Polyadic decomposition), 这种方法为每个实体学习出一个头实体嵌入和一个尾实体嵌入, 而头尾嵌入是独立的。这导致了该方法在补全上性能较差。Simple 基于 CP 方法提出了一种张量分解方法, 解决其训练中头尾无关的问题。每个实体  $e$  具有两个嵌入  $h_e$  和  $t_e$ , 分别表示其在头实体和尾实体中的表示, 每个关系有两个表示。

a) 对于自反的关系  $r$  和  $r^{-1}$ , 可以通过绑定  $v_r$  和  $v_{r^{-1}}$  两个参数 b) 对于反对称关系, 可以将  $v_{r^{-1}}$  绑定成  $-v_r$  c) 对于关系  $r_1, r_2$  使得  $(e_i, r_1, e_j)$  和  $(e_j, r_2, e_i)$  总是同时成立, 可以绑定两个关系的  $v_r$  和  $v_{r^{-1}}$  两个参数

理论分析 Simple 模型的完全表达能力: 当嵌入维数充分大时, Simple 能够完全表示 ground truth。

Representing Asymmetric Relations: 第一种方法对关系矩阵施加了严格的约束, 并对应用于实体嵌入的转换类型施加了严格的限制

后一种方法的缺点是参数个数随关系个数的二次增长, 经常导致过拟合, 特别是对训练三元组数量较少的关系。

Tucker: 通过拥有一个不对称的关系无关的核心张量, 这使得关系之间的知识共享成为可能。它能够对实体嵌入执行所有可能的线性变换, 即旋转、反射或拉伸, 因此能够建模非对称性。不管建模特定关系需要什么样的转换, Tucker 都能够从数据中学习, 而不是通过显式地限制关系矩阵。