知识图谱表示学习-Trans系列算法介绍

**背景介绍**

知识图谱的成功应用引起了业界的广泛关注，并迅速的成为了各行各业的研究热点。基于知识图谱的关系抽取、基于知识图谱的知识推理和基于知识图谱的问答系统的相关论文迅速占据了顶会和顶刊的版面，其中对于知识图谱的表示学习更是为这些任务提供了较底层的支持。表示学习的核心思想即挖掘数据的关键性特征，通过设计映射函数将能够充分表征原始数据的信息以一个低维的向量来表示。以网络的表示学习为例简单的说明一下，如图1所示。

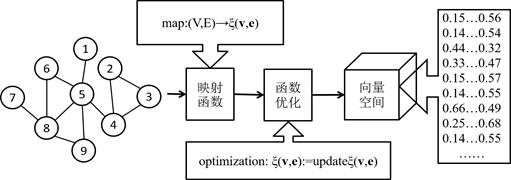


图1 网络表示学习核心思想

网络的表示学习即在原始的网络数据中分析能够度量两个节点相似程度的属性特征，目前经典的LINE、node2vec等方法多是通过假定直接相连的节点更相似、共享更多共同邻居节点和拥有相似结构的节点具有相似性来设计一种将网络空间的节点映射到低维向量空间的表示函数，并通过遍历网络、负采样等方法对表示函数进行优化，从而得到一个可以基本保持原始数据特征的向量空间，即网络空间中相似的两个节点在向量空间中具有相似性（一般使用欧式距离或者cosine相似度进行度量）。

知识图谱的表示学习即将知识图谱构建成一个（头实体，关系，尾实体）的三元组形式，通过目标函数将实体和关系分别以低维的向量来表示。Trans系列的知识图谱表示方法均采用同样的函数思想，即，其中h, t分别表示知识图谱中的头实体和尾实体的向量表示，r表示为关系的向量表示。Trans方法主要有TransE、TransH、TransR、CtransR、TransD、TransA以及TransG等。

**Trans系列算法**

**TransE**算法是早期知识图谱表示学习的典型算法。该算法设计的动机源于作者对于知识库的分层认知和自然传递表示的认识。作者认为知识图谱中的实体和关系是一层一层的进行组织的，同时实体可以通过某种传递方式得到其他实体的信息，从而可以用于对实体进行表示。同时，2013年提出了词向量的概念，并在文本数据上取得了很大的进步，而词向量就是在语料文档的词向量空间寻找相同性质的词，即C(king) – C(queen) ≈ C(man) – C(woman)。又，词向量空间中的传递是存在不变性的。受到这两点的启发，作者提出了TransE算法。

给定知识图谱其中h为头实体集合，t为尾实体集合，l为关系集合。然后将实体和关系在各自的空间中（实体空间和关系空间）表示为向量，使知识图谱中的每个三元组的实体向量和关系向量满足：即如果三元组是正确的，则尾实体向量应该与头实体向量和关系向量的加和更为接近；反之，如果三元组是错误的，则尾实体向量应该与头实体向量和关系向量的加和更为远离。基于此，TransE算法的目标函数设计为: (1)

其中，[·]+表示max(0,x)，d(·)表示一种衡量三元组能量(energy)的距离函数，可以是L1范数也可以是L2范数。Γ表示的是三元组正例和负例的间隔，为超参，类似于SVM分类算法中的“间隔”。从目标函数中可看到需要负例(h’,l,t’)来进行函数的优化。但是众所周知，知识图谱中存储的均为正例，所以就需要人为的构建负例。TransE算法使用的是如下所示的负例构建方法。

(2)

即从知识图谱中随机抽取其他头实体和尾实体来替换当前三元组中头实体或者尾实体，但是为了防止替换之后的三元组也是正例，TransE算法的限制是替换过程不能同时间进行。

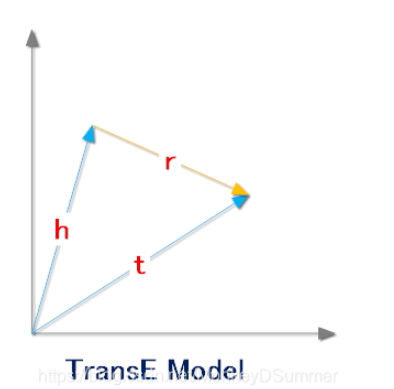
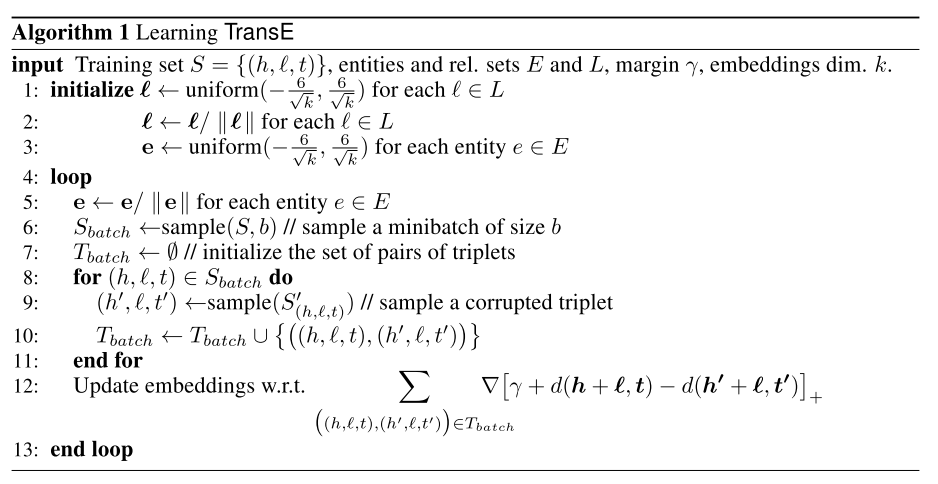


图2 transE算法模型

模型优化过程为最小化目标函数L，即时d(h+l,t)的值小，d(h’+l,t’)的值大，同时使用随机梯度下降（SGD）加速优化过程。如算法1所示：



第一步：随机初始化头实体、尾实体和关系向量，但是对随机初始化的向量做归一化处理；第二步：对数据集中的三元组抽样，并对抽样的三元组进行实体替换，形成负例；第三步：优化目标函数，得到实体和关系的向量表示。从算法描述中可以看出，TransE对于出现在一个既出现在头部又出现在尾部的实体的向量表示是没有区分的。

**transE**算法对知识图谱的表示学习提供了很好的研究思路，但是它只能很好的解决知识图谱中1对1的关系表示，而对于1对多，多对1和多对多的关系类型就无法很好的进行表示。同时对于有自反映射的关系也是无法解决的。比如，从TransE算法的目标函数中可以得到，当三元组为黄金三元组是, 但如果r是自反映射，则r=0，即h=t。如此，在多对1和1对多的场景中，有。因此忽略了实体的分布表示。

**TransH**主要对知识图谱中实体的分布表示加以利用，将不同的实体根据不同的关系映射到超平面上进行向量表示，从而利用了实体的分布表示。TransH首先将知识图谱中的某一种关系类型映射与一个超平面中，根据超平面中的关系，将不同的实体通过映射矩阵投影到该超平面上，从而使得每一个实体都有自己的分布表示，如图3所示。

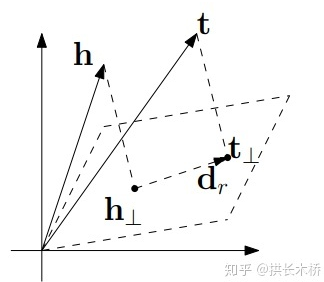


图3 transH算法模型

这样做的目的，主要是因为TransE模型在反射，一对多，多对一等关系中处理的效果并不好。举个例子，在一个对多的关系中，头结点由一个向量表示，它在一个关系中指向多个不同的节点，而这个关系也是由一个vector表示的；理想情况下，如果h + r - t = 0，这就会导致指向的多个节点所计算出来的向量都是相同的值，这明显是有问题的。而且同时，这也造成一个节点在连接不同relations中时，都使用一个相同的vector表示。而在TransH中，为了改善上述问题，引入超平面来代替原有关系向量，从而使得同一个节点在不同关系超平面的向量表示不尽相同。

假设h,r和t是某一个三元组中的一个实例，将r投影到超平面上得到关系投影dr, 根据关系r的投影dr, 将h和t同样映射于该超平面上，则实体在该平面上是具有其独立的分布表示的。

具体的算法实现，对于一个三元组，我们首先需要将h和t映射到我们的超平面上，从而得到映射向量和, 具体公式如下：

得到投影之后我们就可以根据下面的score function来求得三元组的差值:

论文中设置的得分函数为, 通过将关系映射到超平面上，则从向量的投影公式可以得到头实体和尾实体的一个投影表示。因此得分函数可以重新被修改，具体过程如下。

在优化目标函数L的过程中，与TransE一样，最小化L, 使得正例三元组的值小，负例三元组的值大。同时，为了解决TransE算法的负采样问题，TransH提出了一种新的负例生成方法，用以解决1对多和多对1的情况。

基本原则是对于1对多的三元组，给予头实体更多的机会，对于多对一的三元组，给予尾实体更多的机会。所以论文使用一个伯努利分布来分配头尾实体的替换。tph / tph+hpt （分配头实体）& hpt / tph+hpt（分配为实体）。其中，tph表示每个尾实体的平均头实体数， hpt 表示每个头实体的平均为实体数。 TransH存在的问题是虽然考虑了实体自身的分布表示，但是对实体的关系的表示依然是在同一空间（超平面空间）中进行的。可是本质上，实体和关系是相互区别的。

**总结**

Trans系列的算法在知识图谱的表示学习上逐渐完善，但是同样存在很多未能解决的点，比如如何解决多元关系的预测问题，如何结合实体本身的属性信息等等。