Knowledge Embedding Based Graph Convolutional Network

WWW 2021

2021/5/18

1 Introduction

Motivation：

1. 传统的GCN方法主要假设在同质图上进行学习，忽略了KG中的relation蕴含的丰富的信息。
2. 传统的KGE方法没有考虑graph的结构信息
3. 将GCN和KGE结合的方法比如VR-GCN，COMPGCN等，在学习relation embedding的时候没有考虑entity embedding对relation embedding的影响

1 Introduction

Method：

为了解决上面的问题，提出了KE-GCN（Knowledge Embedding based Graph Convolution Network）

- 结合KGE模型，基于图卷积操作同时学习entity和relation embedding。
- 引入了得分函数，认为认为GCN传播的邻居信息是得分函数对于中心节点的梯度。

2 Method

2.1 重新审视GCN

GCN的原始公式

$$\mathrm{m}_v^{l+1} = \sum_{u \in \mathcal{N}(v)} \mathrm{h}_u^l$$

$$\mathrm{h}_v^{l+1} = \sigma(W^l(\mathrm{m}_v^{l+1} + \mathrm{h}_v^l))$$

对于上面的式子,引入一个得分函数f，该得分函数计算edge存在的score，对于已经存在的edge输出较大的值；对于不存在的边输出较小的值。

如果假设得分函数f为： $f(h_u, h_v) = h_u^T h_v$

那么计算的消息h_u能够看做f是对中心实体v的梯度，那么所有h_u加起来就成为下面的形式：

2 Method

$$\mathrm{m}_v^{l+1} = \sum_{u \in \mathcal{N}(v)} \mathrm{h}_u^l \quad \Rightarrow \quad \mathrm{m}_v^{l+1} = \sum_{u \in \mathcal{N}(v)} \frac{\partial f(\mathrm{h}_u^l, \mathrm{h}_v^l)}{\partial \mathrm{h}_v^l} = \frac{\partial (\sum_{u \in \mathcal{N}(v)} f(\mathrm{h}_u^l, \mathrm{h}_v^l))}{\partial \mathrm{h}_v^l}$$

$$\mathrm{h}_v^{l+1} = \sigma(W^l(\mathrm{m}_v^{l+1} + \mathrm{h}_v^l))$$

此时对于$h_v + m_v$看做是learning rate为1，对h_v的梯度提升；目的是使scoring function f的值最大。

通过修改为上面的形式，能够看到，它从新的角度说明了GCN做了什么，邻居信息是如何提供给中心节点的，是如何帮助中心节点获得更好的表示的。

2 Method

2.2 KE-GCN

基于前面的分析，作者提出了新的框架KE-GCN，核心两部分，更新实体表示以及更新关系表示：

$$\mathrm{m}_v^{l+1} = \sum_{(u,r)\in\mathcal{N}_{\text{in}}(v)} W_r^l \frac{\partial f_{\text{in}}(\mathrm{h}_u^l, \mathrm{h}_r^l, \mathrm{h}_v^l)}{\partial \mathrm{h}_v^l} + \sum_{(u,r)\in\mathcal{N}_{\text{out}}(v)} W_r^l \frac{\partial f_{\text{out}}(\mathrm{h}_v^l, \mathrm{h}_r^l, \mathrm{h}_u^l)}{\partial \mathrm{h}_v^l}$$

$$\mathrm{h}_v^{l+1} = \sigma_{\text{ent}}(\mathrm{m}_v^{l+1} + W_0^l \mathrm{h}_v^l)$$

更新实体

$$\mathrm{m}_r^{l+1} = \sum_{(u,v)\in\mathcal{N}(r)} \frac{\partial f_r(\mathrm{h}_u^l, \mathrm{h}_r^l, \mathrm{h}_v^l)}{\partial \mathrm{h}_r^l}$$

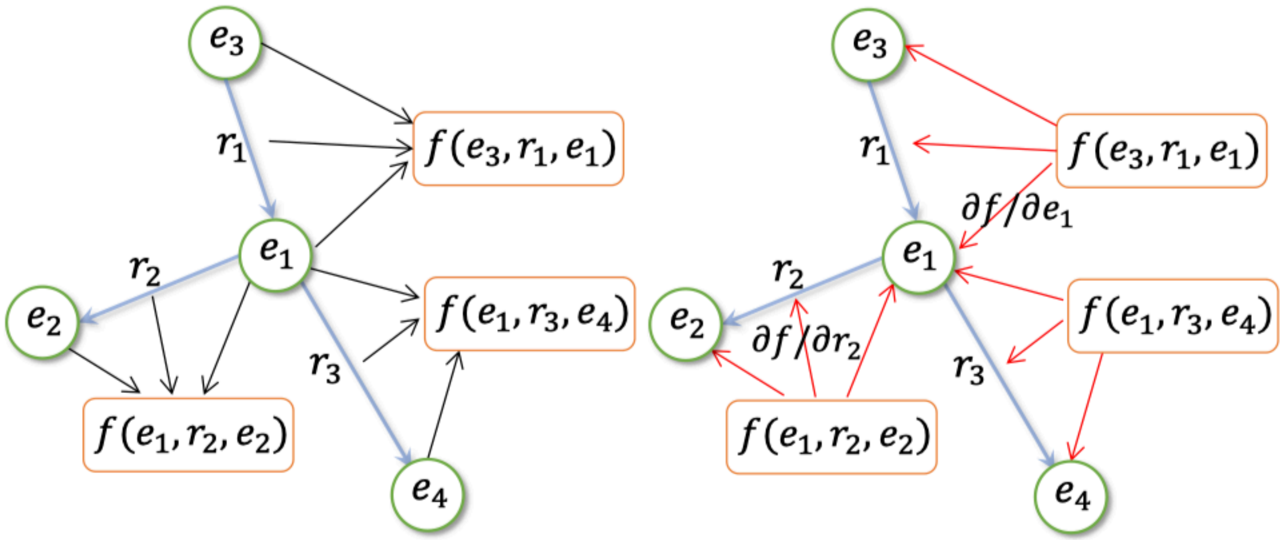
$$\mathrm{h}_r^{l+1} = \sigma_{\text{rel}}(W_{\text{rel}}^l(\mathrm{m}_r^{l+1} + \mathrm{h}_r^l))$$

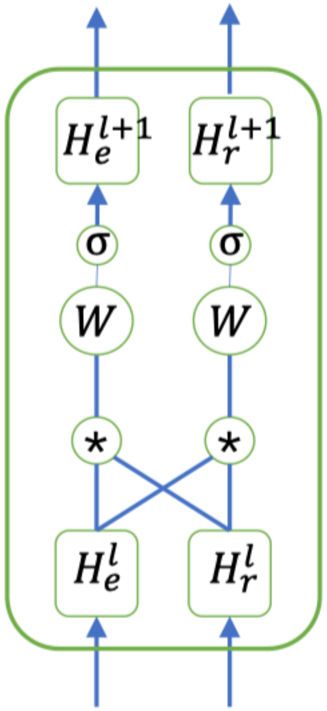
更新关系

2 Method

2.2 KE-GCN

使用图示说明





KE-GCN

消息传递过程；

首先计算得分函数，然后分别求导得到消息

消息聚合过程；

对于实体和关系都有一个聚合函数

3 Experiment

在实验的时候，对于KE-GCN，主要是引入不同的得分函数f，该得分函数使用不同的KGE方法，并且为了简化模型，对于in，out，self-loop都使用了相同的W和相同的得分函数f 。

使用了一系列的KGE方法：

- TransE [3]: For $\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v \in \mathbb{R}^d$,

$$f(\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v) = -\|\mathrm{h}_u + \mathrm{h}_r - \mathrm{h}_v\|_2^2. \tag{16}$$

- DistMult [41]: For $\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v \in \mathbb{R}^d$.

$$f(\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v) = \mathrm{h}_u^T \operatorname{diag}(\mathrm{h}_r)\mathrm{h}_v. \tag{17}$$

- TransH [36]: For $\mathrm{h}_u, \mathrm{h}_v \in \mathbb{R}^d, \mathrm{h}_r \in \mathbb{R}^{2d}$, and $\mathrm{h}_{r1}, \mathrm{h}_{r2} \in \mathbb{R}^d$,

$$f(\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v) = -\|\mathrm{h}_u' + \mathrm{h}_{r2} - \mathrm{h}_v'\|_2^2, \tag{18}$$

$$\mathrm{h}_u' = \mathrm{h}_u - \mathrm{h}_{r1}^T \mathrm{h}_u \mathrm{h}_{r1}, \tag{19}$$

$$\mathrm{h}_v' = \mathrm{h}_v - \mathrm{h}_{r1}^T \mathrm{h}_v \mathrm{h}_{r1}, \tag{20}$$

$$\mathrm{h}_r = [\mathrm{h}_{r1}; \mathrm{h}_{r2}], \tag{21}$$

- TransD [10]: For $\mathrm{h}_u, \mathrm{h}_v, \mathrm{h}_r \in \mathbb{R}^{2d}$, and $\mathrm{h}_{u1}, \mathrm{h}_{u2}, \mathrm{h}_{v1}, \mathrm{h}_{v2}, \mathrm{h}_{r1}, \mathrm{h}_{r2} \in \mathbb{R}^d$,

$$f(\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v) = -\|\mathrm{h}_u' + \mathrm{h}_{r2} - \mathrm{h}_v'\|_2^2, \tag{22}$$

$$\mathrm{h}_u' = \mathrm{h}_{u1} + \mathrm{h}_{u2}^T \mathrm{h}_{u1} \mathrm{h}_{r1}, \tag{23}$$

$$\mathrm{h}_v' = \mathrm{h}_{v1} - \mathrm{h}_{v2}^T \mathrm{h}_{v1} \mathrm{h}_{r1}, \tag{24}$$

$$\mathrm{h}_u = [\mathrm{h}_{u1}; \mathrm{h}_{u2}], \mathrm{h}_v = [\mathrm{h}_{v1}; \mathrm{h}_{v2}], \mathrm{h}_r = [\mathrm{h}_{r1}; \mathrm{h}_{r2}], \tag{25}$$

where $[\cdot;\cdot]$ means concatenation of two vectors.

- RotatE [24]: For $\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v \in \mathbb{C}^d$,

$$f(\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v) = -\|\mathrm{h}_u \circ \mathrm{h}_r - \mathrm{h}_v\|_2^2, \tag{26}$$

where $\circ$ denotes element-wise product and the modulus of any element in h_r is 1, i.e. $|\mathrm{h}_r[i]| = 1\ \forall i \in \{1, 2, \cdots, d\}$. The norm of complex vector is defined as $\|\mathrm{v}\|_p = \sqrt[p]{\sum |\mathrm{v}_i|^p}$.

- QuatE [47]: For $\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v \in \mathbb{H}^d$,

$$f(\mathrm{h}_u, \mathrm{h}_r, \mathrm{h}_v) = \mathrm{h}_u \otimes \mathrm{h}_r \cdot \mathrm{h}_v, \tag{27}$$

3 Experiment

预测任务：

1. Knowledge Graph Alignment
2. Knowledge Graph Entity Classification

知识图谱对齐，主要是匹配两个不同知识图谱中的实体和关系；

知识图谱实体分类，包括多分类和多标签分类

Datasets		#Entities	#Relations	#Triplets
$DBP_{ZH\text{-}EN}$	Chinese	66,469	2,830	153,929
	English	98,125	2,317	237,674
$DBP_{JA\text{-}EN}$	Japanese	65,744	2,043	164,373
	English	95,680	2,096	233,319
$DBP_{FR\text{-}EN}$	French	66,858	1,379	192,191
	English	105,889	2,209	278,590

图谱对齐

Datasets	AM	WN	FB15K
#Entities	1,666,764	40,551	14,904
#Relations	133	18	1,341
#Triplets	5,988,321	145,966	579,654
#Labeled	1,000	31,943	13,445
#Classes	11	24	50

实体分类

3 Experiment

图谱对齐实验结果：

Models	$DBP_{ZH\text{-}EN}$			$DBP_{JA\text{-}EN}$			$DBP_{FR\text{-}EN}$		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
MTransE*[7]	0.364	30.8	61.4	0.349	27.9	57.5	0.335	24.4	55.6
IPTransE*[49]	0.516	40.6	73.5	0.474	36.7	69.3	0.451	33.3	68.5
JAPE*[25]	0.490	41.2	74.5	0.476	36.3	68.5	0.430	32.4	66.7
AlignE*[26]	0.581	47.2	79.2	0.563	44.8	78.9	0.599	48.1	82.4
GCN-Align*[35]	0.549	41.3	74.4	0.546	39.9	74.5	0.532	37.3	74.5
MuGCN*[6]	0.611	49.4	**84.4**	0.621	50.1	**85.7**	0.621	49.5	87.0
AliNet*[27]	0.628	53.9	82.6	0.645	54.9	83.1	0.657	55.2	85.2
R-GCN*[21]	0.564	46.3	73.4	0.571	47.1	75.4	0.570	46.9	75.8
W-GCN [22]	0.553	43.6	73.8	0.554	41.2	74.7	0.541	39.8	74.4
VR-GCN [42]	0.501	38.0	73.3	0.470	35.2	72.2	0.495	36.1	75.1
KBGAT [16]	0.582	48.0	77.3	0.582	47.6	77.7	0.593	47.4	80.9
CompGCN[30]	0.605	49.4	81.2	0.614	50.4	82.2	0.625	50.5	85.0
CompGCN†	0.628	52.8	81.1	0.629	52.8	81.5	0.641	52.6	85.4
KE-GCN	**0.664**	**56.2**	84.2	**0.670**	**57.0**	85.2	**0.683**	**57.2**	**88.5**

3 Experiment

实体分类实验结果：

Models	AM	WN
GCN	86.2 ± 1.4	53.4 ± 0.2
R-GCN	89.3*	55.1 ± 0.6
W-GCN	90.2 ± 0.9*	54.2 ± 0.5
KBGAT	85.7 ± 1.7	53.7 ± 1.1
CompGCN	90.6 ± 0.2*	55.9 ± 0.4
KE-GCN	**91.2 ± 0.2**	**57.8 ± 0.5**

Models	P@1	P@5	N@5
GCN	86.1 ± 0.3	69.0 ± 0.3	82.7 ± 0.2
R-GCN	91.7 ± 0.6	73.0 ± 0.4	89.5 ± 0.6
W-GCN	91.2 ± 0.6	72.8 ± 0.3	88.6 ± 0.5
KBGAT	90.5 ± 0.7	72.4 ± 0.8	87.5 ± 0.8
CompGCN	92.5 ± 0.1	74.0 ± 0.3	90.1 ± 0.2
KE-GCN	**94.3 ± 0.2**	**74.7 ± 0.2**	**91.6 ± 0.2**

3 Experiment

对于KE-GCN使用不同得分函数的实验：

KE-GCN (X)	MRR	H@1	H@10
X = TransE	0.669 ± 0.002	55.9 ± 0.2	87.5 ± 0.2
X = TransH	0.673 ± 0.002	56.1 ± 0.2	87.7 ± 0.2
X = DistMult	0.640 ± 0.002	52.4 ± 0.2	84.7 ± 0.2
X = TransD	0.660 ± 0.002	54.2 ± 0.2	87.6 ± 0.1
X = RotatE	0.673 ± 0.002	56.0 ± 0.3	88.2 ± 0.2
X = QuatE	**0.683 ± 0.002**	**57.2 ± 0.3**	**88.5 ± 0.2**

在实体对齐任务上，性能越好的得分函数获得了越好的效果；

KE-GCN (X)	AM	WN
X = TransE	**91.2 ± 0.2**	**57.8 ± 0.5**
X = TransH	90.5 ± 0.3	57.4 ± 0.3
X = DistMult	89.5 ± 0.4	56.4 ± 0.1
X = TransD	90.1 ± 0.2	57.1 ± 0.2
X = RotatE	90.6 ± 0.4	56.6 ± 0.3
X = QuatE	91.0 ± 0.4	56.9 ± 0.3

但是在实体分类任务上，简单的TransE获得了最好的效果；