

2.21	PME : Projected Metric Embedding on Heterogeneous Networks for Link Prediction	KDD 2018	
2.21	Relation Structure-Aware Heterogeneous Information Network Embedding	AAAI 2019	Pytorch

两篇论文在metric learning和translation embedding的角度对异构信息网络中的节点进行学习，都受到了knowledge graph的模型如TransE，TransR的影响，所以在这里一起来进行对比说明。

PME : Projected Metric Embedding on Heterogeneous Networks for Link Prediction

待解决的问题：

1. 目前大多数异构信息网络(HIN)对于点之间相似度的衡量方式，都是在低维空间使两个点的embedding的**内积(dot product)** 尽可能的大。这种建模方式仅能考虑到一阶关系（first-order proximity），这点在node2vec中也提到。
2. 相比于同构信息网络，异构信息网络中包含**多种relationship**，每种relationship有着不同的**语义信息**。
3. 同时relationship的种类分布非常**不均匀**。

解决的方法：

1. 使用**metric learning**（具体可参见论文collaborative metric learning，它具有triangle inequality特性）来同时捕捉一阶关系和二阶关系（second-order proximity）。
2. 在**object space**学习node的embedding，在**relation space**学习relation的embedding。计算时，先将node embedding从object space转移到relation space，然后计算proximity。
3. 提出loss-aware 自适应采样方法来进行模型优化。

模型的动机：

1. 相比于同构网络的embedding，异构网络中节点之间的proximity不仅仅指两个节点在embedding space的距离，他同时也会受到relation中所包含关系的影响。
2. dot product仅能够保证一阶关系，而metric learning能够更好同时保存一阶关系和二阶关系。

3. 由于metric learning直接应用会存在ill-posed algebraic的问题，所以不能直接应用。同时我们还要考虑到异构网络中存在不同的relation，这点也需要建模。
4. 以往异构网络中，对于不同种类的relation比例差距悬殊的问题，有人提出对每一种relation进行等比例采样，但这会造成有的relation被欠采样，有的过采样，并且不同relation的难度不同，需要采样的数量也不同。

模型：

1. 学习embedding的loss如下，

$$L_r = \sum_{(v_i, v_j) \in D_r} \sum_{(v_i, v_k) \notin D_r} [m + f_r(v_i, v_j)^2 - f_r(v_i, v_k)^2]_+ \quad (4)$$

$$L = \sum_{r \in \mathcal{R}} \sum_{(v_i, v_j) \in D_r} \sum_{(v_i, v_k) \notin D_r} [m + f_r(v_i, v_j)^2 - f_r(v_i, v_k)^2]_+ \quad (5)$$

其中，

$$f_r(v_i, v_j) = w_{ij} \|\mathbf{M}_r \mathbf{v}_i - \mathbf{M}_r \mathbf{v}_j\|, r \in \mathcal{R} \quad (3)$$

可以看出，上述loss的目的是让不同的点在某一种relation space中尽可能地接近，同时是学到的embedding保留一阶和二阶特性。需要学习的参数为node embedding v_i 和从object space映射到不同relation space的映射矩阵 M_r 。

2. 上式中，所有负样本都加入训练集，会导致复杂度急剧上升，在这里采用双向负采样（Bidirectional Negative Sampling Strategy），所以loss修改如下，

$$\begin{aligned} O = & \sum_{r \in \mathcal{R}} \sum_{(v_i, v_j) \in D_r} \\ & \left(\sum_{k=1}^K E_{v_k \sim p_n(v)} [m + f_r(v_i, v_j)^2 - f_r(v_i, v_k)^2]_+ \right. \\ & \left. + \sum_{k=1}^K E_{v_k \sim p_n(v)} [m + f_r(v_i, v_j)^2 - f_r(v_k, v_j)^2]_+ \right) \end{aligned} \quad (7)$$

3. 对于每个epoch，我们会把每个种类的网络的loss记录下来，如下 $L = (l_1, l_2, l_3, \dots, l_{|\mathcal{R}|})$ 然后根据目标种类 r 的loss所占的比例，来确定对该种类 r 的edge采样出多少的比例。这样为根据loss来自适应的调整采样策略（**Loss-aware Adaptive Positive Sampling Strategy**）。
4. 最终整体的算法流程为，

Algorithm 1 Training PME model

Input: A heterogeneous network $G(V, E, W, \mathcal{R})$, number of stochastic gradient steps, N , number of negative samples for each positive sample, K ;

Output: Embeddings for network vertices and relation-specific projection matrix. (i.e., \mathbf{v}, M_r);

```
1:  $iter \leftarrow 0$  ;
2: while  $iter < N$  do
3:   if  $iter = 0$  then
4:     Initialize the positive sampling probability as proportional to the original link distribution from  $G$ ;
5:   else
6:     Sample  $M$  positive examples based on adaptive positive sampling strategy;
7:   End if
8:   For each sampled positive edge, sample  $K$  negative vertices from both sides of the edge;
9:   Compute gradients and update parameters;
10:  Censor the norm of  $\mathbf{v}$  and projection matrix  $M_r$ ;
11:  Compute relation-specific subgraph loss, and update the positive sampling probability;
12:   $iter \leftarrow iter + 1$  ;
13: end
```

实验:

实验采用了来自五个州的Yelp数据集，点的种类包括用户（User），物品（Business），物品属性（Attribute），物品种类（Category），如Table 1所示。在AZ州的数据集上计算 Hits@K和MRR，结果如Figure 2，和Tabel 3所示。在NV州数据集做link prediction任务，具体为判断当前便是否存在在测试集当中，具体指标使用AUC，结果如Tabel 4所示。

Table 1: Yelp network statistics

State	No. of Edges				No. of Nodes			
	User to User	User to Business	Business to Attributes	Business to Categories	Business	Users	Attributes	Categories
Complete	29,271,479	4,153,150	605,231	527,229	14,4072	1,029,432	81	1,191
NV	4,891,171	1,460,807	106,789	105,358	28,214	428,840	81	1,030
AZ	2,269,462	1,265,915	161,361	162,393	43,492	311,857	81	1,052
ON	465,204	500,812	120,241	84,491	24,507	92,997	66	777
WI	57,593	88,778	14,986	18,479	3,899	25,773	81	678
EDH	25,695	44,631	12,676	11,972	3,539	8,371	72	456

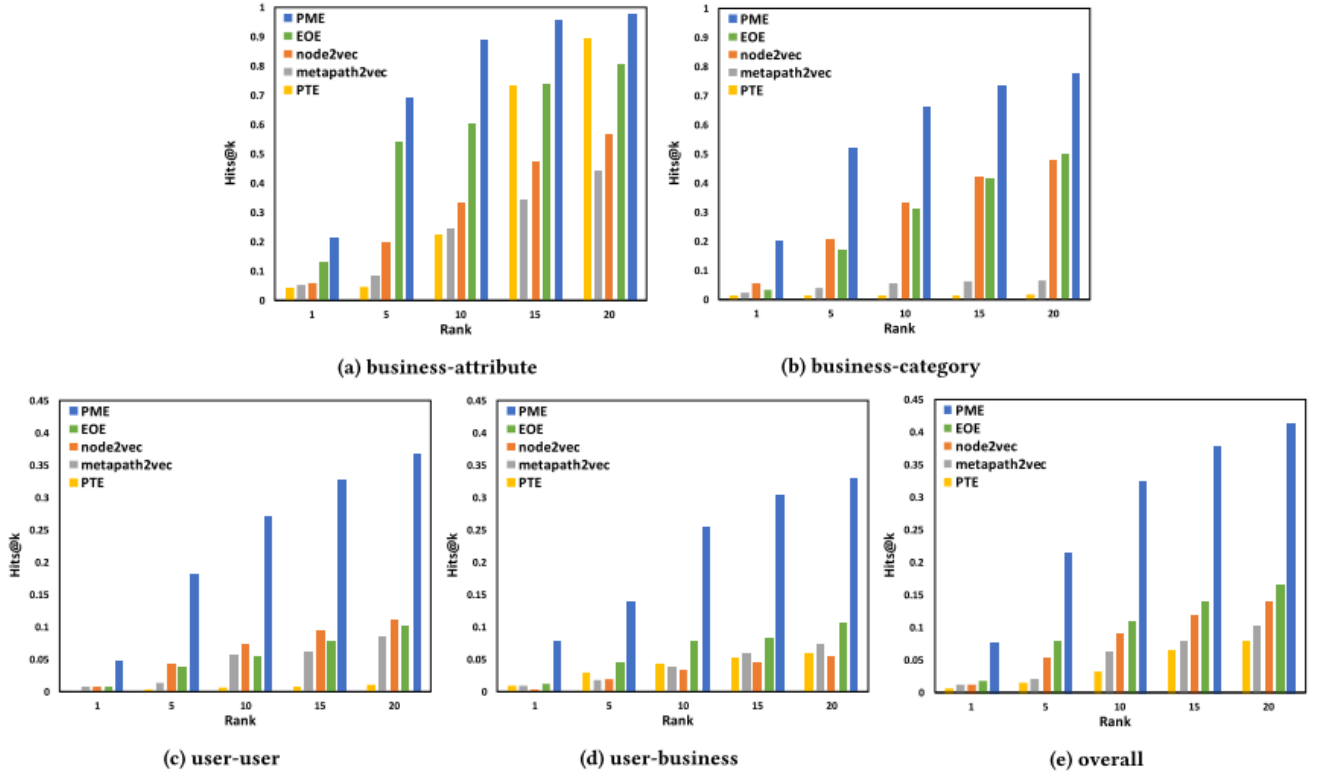


Figure 2: Hit ratio@ top 20, 15, 10, 5, 1

Table 3: Predication accuracy in terms of MRR

	PME	node2vec	PTE	EOE	metapath2vec
Overall	0.1253	0.0396	0.0181	0.0624	0.0098
user-user	0.1249	0.0314	0.0036	0.0260	0.0019
user-business	0.0529	0.0163	0.0219	0.0403	0.0089
business-attribute	0.3701	0.1539	0.1179	0.3059	0.0547
business-category	0.3151	0.1418	0.0321	0.2923	0.0435

Table 4: AUC scores on NV network

	PME	node2vec	PTE	EOE	metapath2vec
Overall	0.9618	0.8789	0.7494	0.8562	0.6232
user-user	0.9672	0.8909	0.6347	0.9033	0.5141
user-business	0.9590	0.8835	0.8615	0.9129	0.8179
business-attribute	0.9376	0.7522	0.8944	0.9201	0.5653
business-category	0.9896	0.9233	0.9652	0.9819	0.7725

总结:

该篇文章整体的贡献点为, 1. metric learning来解决HIN中的二阶关系, 并借用TransR中的映射矩阵来解决metric learning存在的ill-posed algebraic问题, 对于多种relation建立多个relation space. 2. 提出loss-aware adaptive采样方法, 解决了HIN中存在的relation skewed的问题。

但是可能存在的**问题**是, 该篇文章仅仅考虑基础的relation, 另外在HIN中还有常见的composite relations是使用 meta-paths来表示的。比如在DBLP这样的参考文献数据集上, 存在 (A, author, P, paper, C, conference) 这些节点。而像APA (co-author relation), 以及APC (authors write papers published in conferences) 这样包含着丰富的信息的composite relations, 在这篇文章中没有考虑到。

Relation Structure-Aware Heterogeneous Information Network Embedding (oral)

待解决的问题：

1. 异构网络中，存在着很多的relations，不同的relations有着不同的特性，如AP表现的是peer-to-peer，而PC表现的是 one-centered-by-another关系。如何区分不同的relations？
2. 针对不同的relations，目前的模型都采用相同的方法来对他们进行处理。如何区分建模？
3. 如果建立多个模型，如何协调优化？

解决的方法：

1. 根据结构特性定义了两种relations，Affiliation Relations (ARs) 代表one-centered-by-another的结构，而 Interaction Relations (IRs) 代表peer-to-peer的关系。
2. 对于AR，这些点应当有共同的特性，所以直接用节点的欧几里得距离作为proximity。对于IR，将这种关系定义为节点之间的**转移 (translation)**。前者借鉴了collaborative metric learning，后者借鉴了模型TransE。
3. 因为两个模型在数学形式上相似，所以可以一起优化。

数据分析：

不同上篇文章，这篇文章从数据分析入手，并给出两种structural relation的定义。三个数据集整合如Table 1所示。对于一个relation的三元组， $\langle u, r, v \rangle$ ，其中作者定义了一个指标如下，

$$D(r) = \frac{\max[\bar{d}_{t_u}, \bar{d}_{t_v}]}{\min[\bar{d}_{t_u}, \bar{d}_{t_v}]}, \quad (1)$$

该指标由u和v种类的节点的平均 (**degrees**) 度来决定。如果 $D(r)$ 越大，代表由r连接的两类节点的不平衡性越大，越倾向于AR类型，否则倾向IR类型。同时定义了另外一个稀疏度指标如下，

$$S(r) = \frac{N_r}{N_{t_u} \times N_{t_v}}, \quad (2)$$

其中 N_r 代表该种类relation的数量， N_{t_u} 代表头节点所在种类节点的数量，如果数据越稠密，则越倾向于AR，因为是一one-centered-by-another，而IR关系的相对来说应该较为稀疏。

Table 1: Statistics of the three datasets. t_u denotes the type of node u , $\langle u, r, v \rangle$ is a node-relation triple.

Datasets	Nodes	Number of Nodes	Relations ($t_u \sim t_v$)	Number of Relations	Avg. Degree of t_u	Avg. Degree of t_v	Measures $D(r)$ $S(r)$		Relation Category
DBLP	Term (T)	8,811	PC	14,376	1.0	718.8	718.8	0.05	AR
	Paper (P)	14,376	APC	24,495	2.9	2089.7	720.6	0.085	AR
	Author (A)	14,475	AP	41,794	2.8	2.9	1.0	0.0002	IR
	Conference (C)	20	PT	88,683	6.2	10.7	1.7	0.0007	IR
			APT	260,605	18.0	29.6	1.6	0.002	IR
Yelp	User (U)	1,286	BR	2,614	1.0	1307.0	1307.0	0.5	AR
	Service (S)	2	BS	2,614	1.0	1307.0	1307.0	0.5	AR
	Business (B)	2,614	BL	2,614	1.0	290.4	290.4	0.1	AR
	Star Level (L)	9	UB	30,838	23.9	11.8	2.0	0.009	IR
	Reservation (R)	2	BUB	528,332	405.3	405.3	1.0	0.07	IR
AMiner	Paper (P)	127,623	PC	127,623	1.0	1263.6	1264.6	0.01	AR
	Author (A)	164,472	APC	232,659	2.2	3515.6	1598.0	0.01	AR
	Reference (R)	147,251	AP	355,072	2.2	2.8	1.3	0.00002	IR
	Conference (C)	101	PR	392,519	3.1	2.7	1.1	0.00002	IR
			APR	1,084,287	7.1	7.9	1.1	0.00004	IR

模型：

对于AR类型，采用类似于上篇文章PME中的metric learning角度建模，原因除了metric learning能够保留second-order proximities外，metric learning和AR的定义契合，及被该关系连接的节点之间欧式距离要尽量的小。而对于IR类型为何用translation来进行建模，没有更好的说明，只是在模型的数学形式上和metric learning较为接近，容易结合。

则对于AR类型的loss为，

$$L_{EuAR} = \sum_{s \in R_{AR}} \sum_{\langle p, s, q \rangle \in P_{AR}} \sum_{\langle p', s, q' \rangle \in P'_{AR}} \max[0, \gamma + f(p, q) - f(p', q')], \quad (4)$$

$$f(p, q) = w_{pq} \|\mathbf{X}_p - \mathbf{X}_q\|_2^2, \quad (3)$$

而对于IR类型的loss为，

$$L_{TrIR} = \sum_{r \in R_{IR}} \sum_{\langle u, r, v \rangle \in P_{IR}} \sum_{\langle u', r, v' \rangle \in P'_{IR}} \max[0, \gamma + g(u, v) - g(u', v')] \quad (6)$$

$$g(u, v) = w_{uv} \|\mathbf{X}_u + \mathbf{Y}_r - \mathbf{X}_v\|, \quad (5)$$

对于整个模型来说，就是简单的把两部分的loss相加，没有上一篇PME中考虑的更合理，

$$\begin{aligned} L &= L_{EuAR} + L_{TrIR} \quad (7) \\ &= \sum_{s \in R_{AR}} \sum_{\langle p, s, q \rangle \in P_{AR}} \sum_{\langle p', s, q' \rangle \in P'_{AR}} \max[0, \gamma + f(p, q) - f(p', q')] \\ &+ \sum_{r \in R_{IR}} \sum_{\langle u, r, v \rangle \in P_{IR}} \sum_{\langle u', r, v' \rangle \in P'_{IR}} \max[0, \gamma + g(u, v) - g(u', v')] \end{aligned}$$

正负采样的方法也没有上一篇当中有过多的技巧，relation的正采样就直接按照数据集中的比例来进行采样，不考虑relation种类是skewed的情况。而对于负采样，和TransE和上篇文章中PME相同的方法，即双向负采样。

实验：

实验采用Table 1中的数据集，首先看在聚类任务上效果的好坏，具体指标采用NMI，结果在表格2中所示。

Table 2: Performance Evaluation of Node Clustering.

Methods	DBLP	Yelp	AMiner
DeepWalk	0.3884	0.3043	0.5427
LINE-1st	0.2775	0.3103	0.3736
LINE-2nd	0.4675	0.3593	0.3862
PTE	0.3101	0.3527	0.4089
ESim	0.3449	0.2214	0.3409
HIN2Vec	0.4256	0.3657	0.3948
metapath2vec	0.6065	0.3507	0.5586
RHINE	0.7204	0.3882	0.6024

接着看了模型在Link prediction上的效果好坏，具体为判断当前边是否在测试集中。具体指标采用AUC和F1，结果见表3。

Table 3: Performance Evaluation of Link Prediction.

Methods	DBLP (A-A)		DBLP (A-C)		Yelp (U-B)		AMiner (A-A)		AMiner (A-C)	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
DeepWalk	0.9131	0.8246	0.7634	0.7047	0.8476	0.6397	0.9122	0.8471	0.7701	0.7112
LINE-1st	0.8264	0.7233	0.5335	0.6436	0.5084	0.4379	0.6665	0.6274	0.7574	0.6983
LINE-2nd	0.7448	0.6741	0.8340	0.7396	0.7509	0.6809	0.5808	0.4682	0.7899	0.7177
PTE	0.8853	0.8331	0.8843	0.7720	0.8061	0.7043	0.8119	0.7319	0.8442	0.7587
ESim	0.9077	0.8129	0.7736	0.6795	0.6160	0.4051	0.8970	0.8245	0.8089	0.7392
HIN2Vec	0.9160	0.8475	0.8966	0.7892	0.8653	0.7709	0.9141	0.8566	0.8099	0.7282
metapath2vec	0.9153	0.8431	0.8987	0.8012	0.7818	0.5391	0.9111	0.8530	0.8902	0.8125
RHINE	0.9315	0.8664	0.9148	0.8478	0.8762	0.7912	0.9316	0.8664	0.9173	0.8262

另外看了模型在multi-class classification任务上的表现，看学到的节点是否保留有节点种类信息，具体为对已经学习到的节点embedding，训练一个分类器，结果如Table 4所示。

Table 4: Performance Evaluation of Multi-class Classification.

Methods	DBLP		Yelp		AMiner	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
DeepWalk	0.7475	0.7500	0.6723	0.7012	0.9386	0.9512
LINE-1st	0.8091	0.8250	0.4872	0.6639	0.9494	0.9569
LINE-2nd	0.7559	0.7500	0.5304	0.7377	0.9468	0.9491
PTE	0.8852	0.8750	0.5389	0.7342	0.9791	0.9847
ESim	0.8867	0.8750	0.6836	0.7399	0.9910	0.9948
HIN2Vec	0.8631	0.8500	0.6075	0.7361	0.9962	0.9965
metapath2vec	0.8976	0.9000	0.5337	0.7208	0.9934	0.9936
RHINE	0.9344	0.9250	0.7132	0.7572	0.9884	0.9807

另外，为了探讨区分两种relation，并利用metric learning和translation进行建模是否有效，作者进行了ablation study。提出如下三种variants，

- **RHINE_{Eu}** leverages Euclidean distance to embed HINs without distinguishing the relations.
- **RHINE_{Tr}** models all nodes and relations in HINs with translation mechanism, which is just like TransE (Bordes et al. 2013).
- **RHINE_{Re}** leverages Euclidean distance to model IRs while translation mechanism for ARs, reversely.

其实验结果如Figure 2所示,

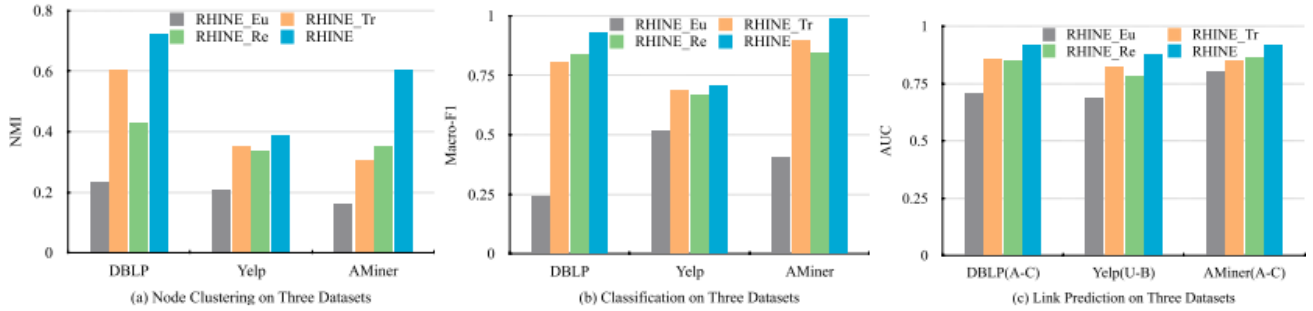


Figure 2: Performance Evaluation of Variant Models.

总结:

总的来说, 作者从分析数据入手, 对于HIN中具有不同structural的relation进行了区分, 并且分别采用不同的方法对不同structural的relation进行建模, 在一定程度上给出了这两种方法的建模motivation。相比于PME, 作者对于两部分的relations的loss结合较为粗糙, 不过作者的重点也不在于此, 没有什么问题。