

阿里电商场景下的大规模异构网络表示学习

原创 岑宇阔 学术头条 2019-05-30

传统的网络(图)表示学习方法一般只针对同构的图，但是实际的图往往都是异构的。只包含异构节点的图的表示学习已经被广泛研究，例如metapath2vec提出了异构的random walk和skip gram。而包含异构边的图的表示学习近来开始被大家所关注，比如MNE给每个点学习多种表示来处理异构边的情形。

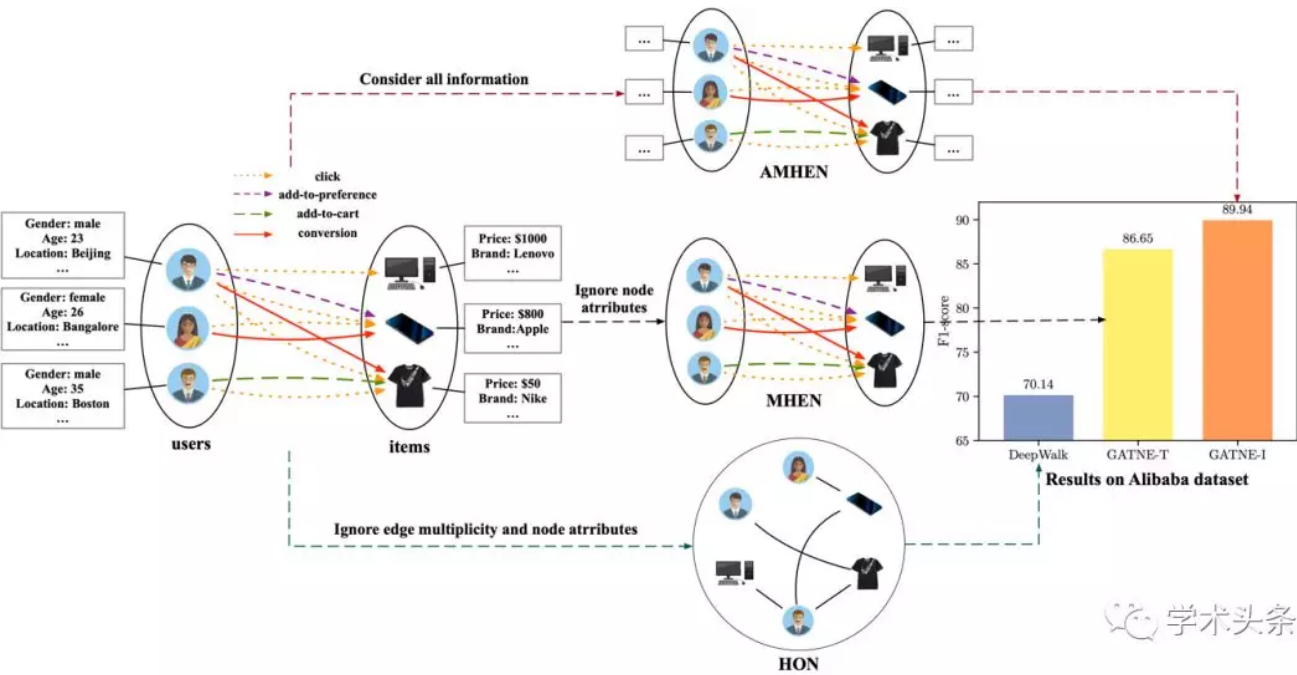
在阿里电商场景下，由用户和商品构成的图就是异构的，不仅包含异构的节点（用户和商品），而且包含异构的边（用户和商品的多种交互行为，比如点击、购买等）。不仅如此，图中的节点还包含着丰富的属性。本文处理的就是这种包含异构节点和异构边的图的表示学习。

根据图结构(同构/异构)以及是否包含节点特征，我们将图分为如下六类：HON, AHON, HEN, AHEN, MHEN, AMHEN。同时我们也在下表中列出了处理各种类型的图的方法(其中GATNE-T/I是我们提出的方法)。

Network Type	Method	Heterogeneity		Attribute
		Node Type	Edge Type	
Homogeneous Network (HON)	DeepWalk [27] LINE [35] node2vec [10] NetMF [29] NetSMF [28]	Single	Single	/
Attributed Homogeneous Network (AHON)	TADW [41] LANE [16] AANE [15] SNE [20] DANE [9] ANRL [44]	Single	Single	Attributed
Heterogeneous Network (HEN)	PTE [34] metapath2vec [7] HERec [31]	Multi	Single	/
Attributed HEN (AHEN)	HNE [3]	Multi	Single	Attributed
Multiplex Heterogeneous Network (MHEN)	PMNE [22] MVE [30] MNE [43] mvn2vec [32]	Single	Multi	/
	GATNE-T	Multi	Multi	
Attributed MHEN (AMHEN)	GATNE-I	Multi	Multi	Attributed

下图是一个带节点属性的异构图的例子。在左侧原始的图中，用户包含了性别、年龄等属性，商品包含了价格、类目等属性。用户与商品之间包含了4种类型的边，分别对应点击、收藏、加

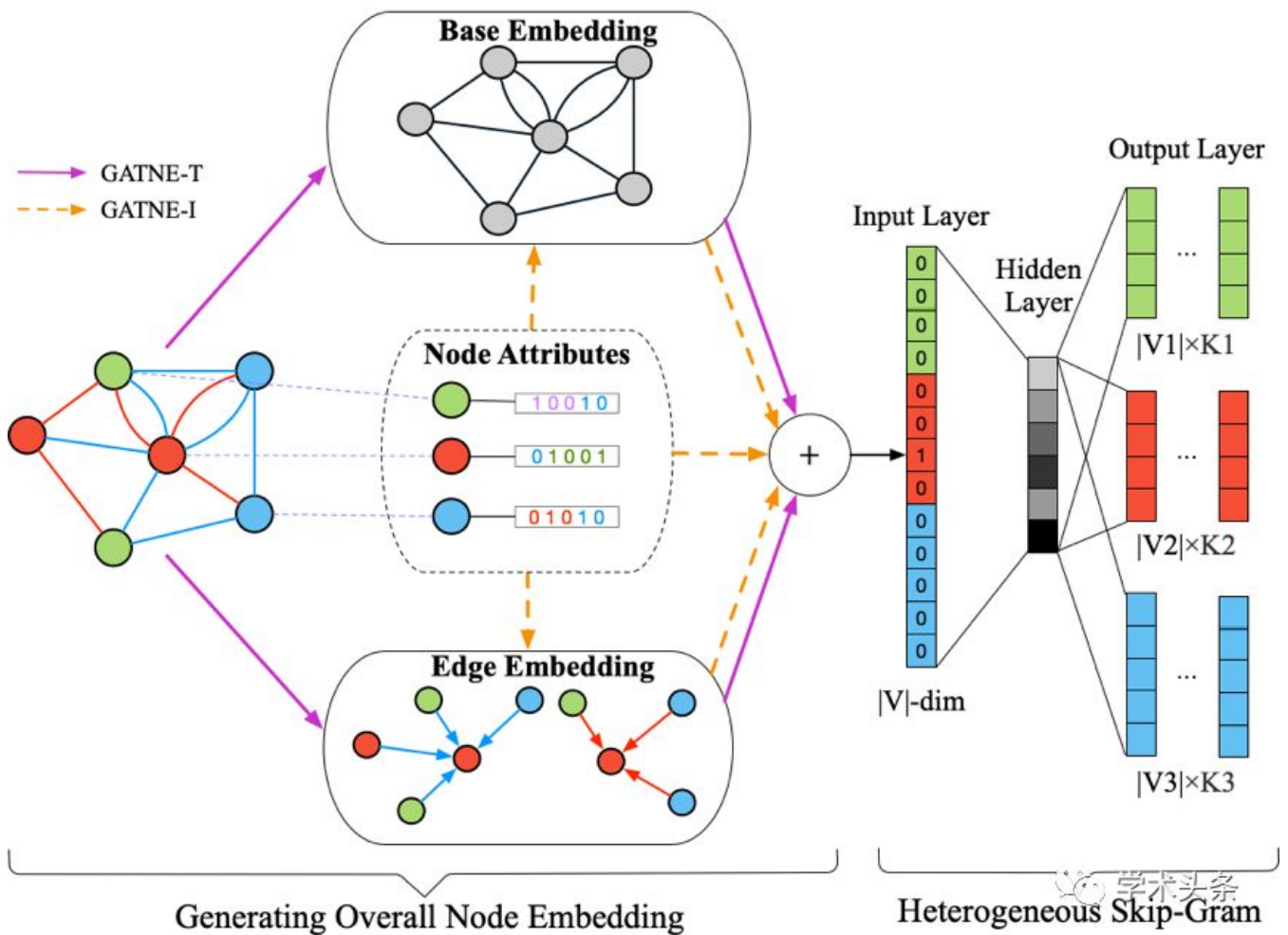
入购物车以及购买行为。



传统的graph embedding算法比如DeepWalk的做法会忽略图中边的类型以及节点的特征，然后转换成一个HON。如果将边的类型考虑进去，那么就得到一个MHEN，能够取得非常明显的效果。此外，如果将节点的属性也同时考虑进去，那么就利用了原图的所有信息，可以得到最好的效果。

模型

由于有多种类型的边(比如点击、购买)，这里我们考虑给每个节点在每种边类型下都学一个表示。比如我们给用户和商品在点击场景下学一种表示，在购买场景下学一种表示。但是这两种表示之间并不是完全独立的，是通过某种机制互相影响的。我们主要考虑的就是如何来建模不同类型的表示之间相互影响的方式。



模型的大致结构如上图所示。对于GATNE-T(T for transductive)来说，每个节点在每种 edge type 下的 embedding 由两个部分组成，分别是 base embedding 和 edge embedding。base embedding 是每个节点在每种 edge type 下共享的，而 edge embedding 是通过相邻节点的 edge embedding 计算得到的。具体来说，类似于 GraphSAGE[3]，这里的 edge embedding 的计算方式如下：

$$\mathbf{u}_{i,r}^{(k)} = \text{aggregator}(\{\mathbf{u}_{j,r}^{(k-1)}, \forall v_j \in \mathcal{N}_{i,r}\})$$

其中 i, j 表示节点编号， r 表示某种 edge type， k 表示第 k 层的 edge embedding ($1 \leq k \leq K$)，aggregator function 可以是 mean aggregator 或者 max-pooling aggregator。我们将第 K 层的 edge embedding 拼起来记为矩阵 \mathbf{U}_i ：

$$\mathbf{U}_i = (\mathbf{u}_{i,1}, \mathbf{u}_{i,2}, \dots, \mathbf{u}_{i,m})$$

其中m表示edge type的数量。由于我们不知道每个节点在每种edge type下的表示之间的关系，所以我们通过self-attention[4]的机制来建模这种相互关系，并得到每种edge type下的表示对于各个edge type的权重：

$$\mathbf{a}_{i,r} = \text{softmax}(\mathbf{w}_r^T \tanh(\mathbf{W}_r \mathbf{U}_i))^T$$

其中 \mathbf{w}_r , \mathbf{W}_r 是模型需要训练得到的参数， $\mathbf{a}_{i,r}$ 是计算得到的权重向量。最后我们就能得到每个节点i在某种edge type r下的向量表示：

$$\mathbf{v}_{i,r} = \mathbf{b}_i + \alpha_r \mathbf{M}_r^T \mathbf{U}_i \mathbf{a}_{i,r}$$

由于实际问题中会遇到冷启动等问题，需要给没有在训练集中出现过的节点也求得embedding。而transductive model不具备这种能力。所以我们引入了节点的特征提出了相应的inductive model，记为GATNE-I。具体来说，原先在GATNE-T中 \mathbf{b}_i 和 $\mathbf{u}_{i,r}(0)$ 都是随机初始化的。但是在GATNE-I，这两个embedding都是基于节点的特征，也就是通过节点的特征经过某种变换(比如线性变换或者神经网络等)得到的。那么节点i在某种edge type r下的向量表示就可以表达成：

$$\mathbf{v}_{i,r} = \mathbf{h}_z(\mathbf{x}_i) + \alpha_r \mathbf{M}_r^T \mathbf{U}_i \mathbf{a}_{i,r} + \beta_r \mathbf{D}_z^T \mathbf{x}_i$$

其中 \mathbf{x}_i 表示节点i的特征。

接下来介绍模型的训练方式。GATNE-T/I模型的训练方式基于meta-path-based random walk和heterogeneous skip gram。具体来说，我们预先设定meta-path scheme (比如User-Item-User)，那么random walk的转移概率即为：

$$p(v_j | v_i, \mathcal{T}) = \begin{cases} \frac{1}{|\mathcal{N}_{i,r} \cap \mathcal{V}_{t+1}|} & (v_i, v_j) \in \mathcal{E}_r, v_j \in \mathcal{V}_{t+1}, \\ 0 & (v_i, v_j) \in \mathcal{E}_r, v_j \notin \mathcal{V}_{t+1}, \\ 0 & (v_i, v_j) \notin \mathcal{E}_r, \end{cases}$$

其中 E_r 表示类型为 r 的边集。给定一个节点与其在某个random walk上的context C ，我们的目标是 minimized 如下的负对数似然函数：

$$-\log P_{\theta}(\{v_j | v_j \in C\} | v_i) = \sum_{v_j \in C} -\log P_{\theta}(v_j | v_i)$$

其中右侧的每一项的概率通过heterogeneous softmax function来计算：

$$P_{\theta}(v_j | v_i) = \frac{\exp(\mathbf{c}_j^T \cdot \mathbf{v}_{i,r})}{\sum_{k \in \mathcal{V}_t} \exp(\mathbf{c}_k^T \cdot \mathbf{v}_{i,r})}$$

其中 \mathbf{c} 表示节点的context embedding。最后我们通过heterogeneous negative sampling来近似负对数似然：

$$E = -\log \sigma(\mathbf{c}_j^T \cdot \mathbf{v}_{i,r}) - \sum_{l=1}^L \mathbb{E}_{v_k \sim P_t(v)} [\log \sigma(-\mathbf{c}_k^T \cdot \mathbf{v}_{i,r})]$$

整体的算法流程如下：

Algorithm 1: GATNE

Input: network $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, embedding dimension d , edge embedding dimension s , learning rate η , negative samples L , coefficient α, β .

Output: overall embeddings $\mathbf{v}_{i,r}$ for all nodes on every edge type r

- 1 Initialize all the model parameters θ .
- 2 Generate random walks on each edge type r as \mathcal{P}_r .
- 3 Generate training samples $\{(v_i, v_j, r)\}$ from random walks \mathcal{P}_r on each edge type r .
- 4 **while** not converged **do**
- 5 **foreach** $(v_i, v_j, r) \in \text{training samples}$ **do**
- 6 Calculate $\mathbf{v}_{i,r}$ using Equation (6) or (13)
- 7 Sample L negative samples and calculate objective function E using Equation (17)
- 8 Update model parameters θ by $\frac{\partial E}{\partial \theta}$.

 学术头条

实验

我们在3个公开数据集Amazon、YouTube、Twitter和阿里数据集上进行了实验，验证了我们所提出的模型的有效性，同时也说明了异构边的信息对如何学到更好的graph embedding能带来较大的帮助。用到的数据集的规模如下（数据集从原始数据集中进行了采样）：

Dataset	# nodes	# edges	# n-types	# e-types
Amazon	10,166	148,865	1	2
YouTube	2,000	1,310,617	1	5
Twitter	10,000	331,899	1	4
Alibaba-S	6,163	17,865	2	4
Alibaba	41,991,048	571,892,183	2	4

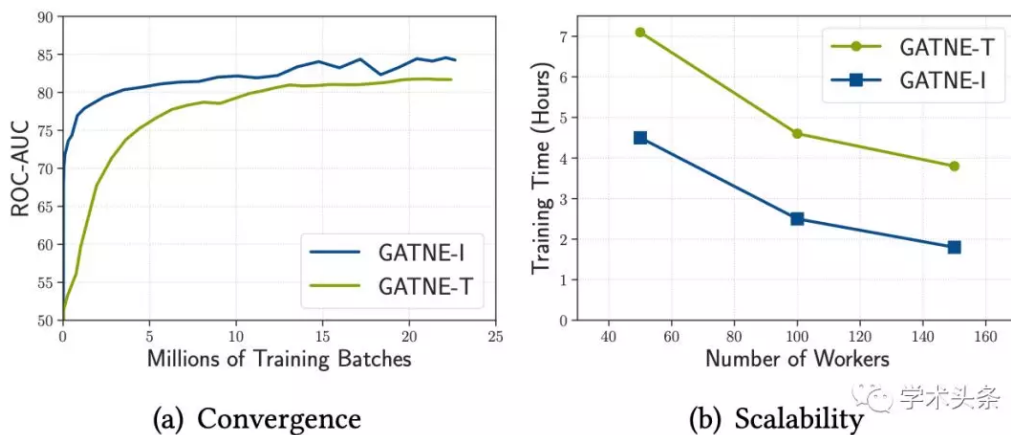
如下表所示，在3个公开数据集和阿里小数据集上，我们提出的GATNE-T/I取得了最好的效果。Amazon数据集上由于商品的特征比较弱，所以GATNE-I的效果会稍差于GATNE-T；而在阿里数据集上，因为节点的特征非常丰富，所以GATNE-I的效果会好于GATNE-T。YouTube和Twitter数据集不包含节点特征，所以我们把DeepWalk跑出来的200维向量作为初始的节点特征。由于DeepWalk生成的特征也只利用了图的结构，没有引入额外的信息，所以两种方法GATNE-T/I的结果差别不大。

	Amazon			YouTube			Twitter			Alibaba-S		
	ROC-AUC	PR-AUC	F1	ROC-AUC	PR-AUC	F1	ROC-AUC	PR-AUC	F1	ROC-AUC	PR-AUC	F1
DeepWalk	94.20	94.03	87.38	71.11	70.04	65.52	69.42	72.58	62.68	59.39	60.62	56.10
node2vec	94.47	94.30	87.88	71.21	70.32	65.36	69.90	73.04	63.12	62.26	63.40	58.49
LINE	81.45	74.97	76.35	64.24	63.25	62.35	62.29	60.88	58.18	53.97	54.65	52.85
metapath2vec	94.15	94.01	87.48	70.98	70.02	65.34	69.35	72.61	62.70	60.94	61.40	58.25
ANRL	71.68	70.30	67.72	75.93	73.21	70.65	70.04	67.16	64.69	58.17	55.94	56.22
PMNE(n)	95.59	95.48	89.37	65.06	63.59	60.85	69.48	72.66	62.88	62.23	63.35	58.74
PMNE(r)	88.38	88.56	79.67	70.61	69.82	65.39	62.91	67.85	56.13	55.29	57.49	53.65
PMNE(c)	93.55	93.46	86.42	68.63	68.22	63.54	67.04	70.23	60.84	51.57	51.78	51.44
MVE	92.98	93.05	87.80	70.39	70.10	65.10	72.62	73.47	67.04	60.24	60.51	57.08
MNE	90.28	91.74	83.25	82.30	82.18	75.03	91.37	91.65	84.32	62.79	63.82	58.74
GATNE-T	97.44	97.05	92.87	84.61	81.93	76.83	92.30	91.77	84.96	66.74	67.55	63.48
GATNE-I	96.25	94.77	91.36	84.47	82.32	76.83	92.04	91.95	84.38	70.87	71.65	65.54

我们实现了3个baseline(DeepWalk, MVE, MNE)和GATNE-T/I的分布式版本，运行在PAI Tensorflow上，并在阿里大数据集上进行了测试。如下表所示，相比于baseline来说，GATNE-I取得了非常显著的提高。

	ROC-AUC	PR-AUC	F1
DeepWalk	65.58	78.13	70.14
MVE	66.32	80.12	72.14
MNE	79.60	93.01	84.86
GATNE-T	81.02	93.39	86.65
GATNE-I	84.20	95.04	89.94

我们对模型的convergence和scalability进行了测试。在阿里大数据集上，相比GATNE-T来说，GATNE-I能够更快地达到比较好的效果。并且，随着模型所使用的worker数量的增加，GATNE-T/I模型的训练时间都能显著降低。



参考文献

- [1] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In KDD'17. ACM, 135–144.
- [2] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. 2018. Scalable Multiplex Network Embedding. In IJCAI'18. 3082–3088.
- [3] Will Hamilton, Zhitaoy Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In NIPS'17. 1024–1034.
- [4] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. ICLR'17.

[关于转载]: 本文为“学术头条”原创文章。转载仅限全文转载并保留文章标题及内容，不得删改、添加内容绕开原创保护，且文章开头必须注明：转自“SciTouTiao”微信公众号。感谢您的合作。