【GNN】GATNE：阿里大规模多元异构属性网络

原创 阿泽crz 阿泽的学习笔记 2020-06-10

收录于话题
#图神经网络

20个

今天学习的是清华大学和达摩院合作的一篇论文《Representation Learning for Attributed Multiplex Heterogeneous Network》，发表于 KDD 2019。

目前很多 Graph Embedding 应用广泛，但大部分都只是同构网络或者是小尺度网络，而真实世界往往大都是数以亿计的不同类型的节点和边，且节点往往包含多种属性。

为此，作者提出了 GATNE 框架用于解决大规模多元异构属性网络（Attributed Multiplex Heterogeneous Network，AMHEN），该框架支持 transductive 和 inductive 的学习范式。此外，作者也进行了理论分析证明了 GATNE 具有良好的表达能力，并通过四种不同的数据集和 A/B 测试验证了模型的性能。

1.Introduction

作者根据网络拓扑类型（同构和异构）和属性特征（边、节点）将图分为六类，并总结了当今的发展，如下表所示：

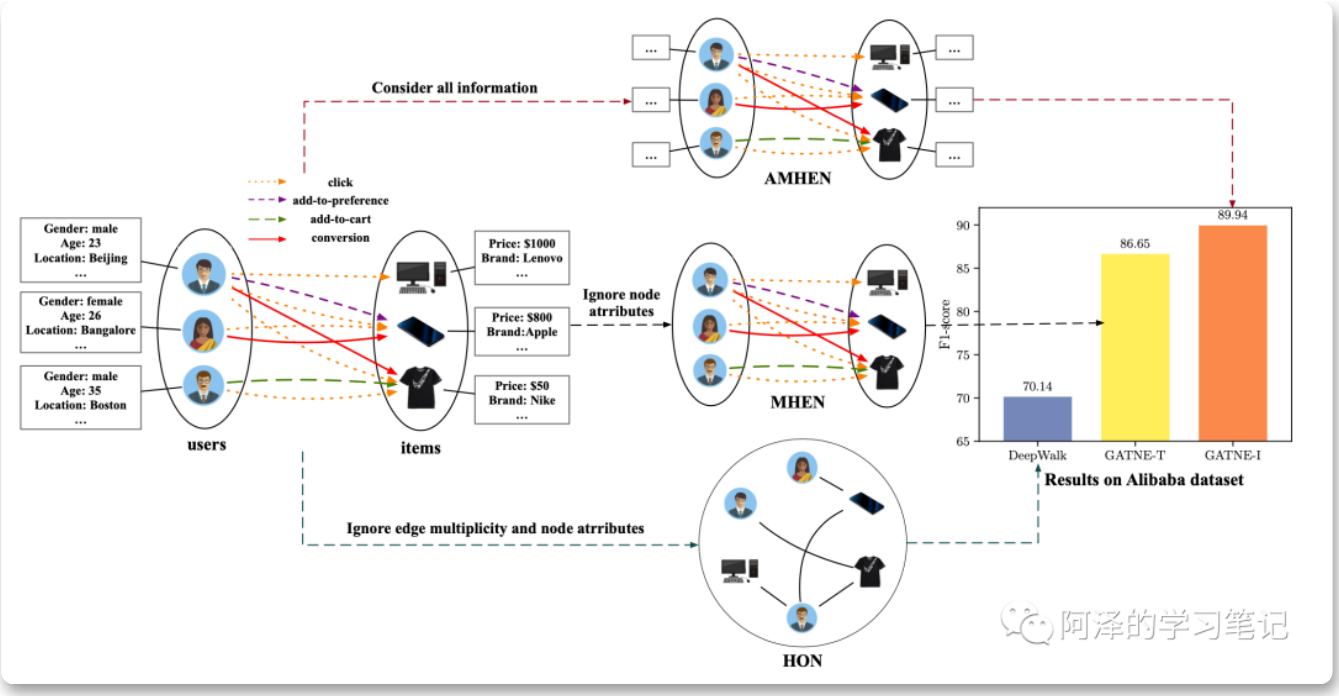
Table 1: The network types handled by different methods.

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

可以看到 AMHEN 的研究是最少的。

本篇论文致力于研究 AMHEN 的表示学习，这种网络的特点在于多种类型的节点可能通过多种类型的边进行关联，并且每个节点都具有不同的属性。

这种结构非常常见，比如作者使用的四种数据集中，超过 15% 的节点对会有超过一种类型的边。举个具体的例子，在电商中，用户可能会对商品具有点击、加购、加收藏等多种交互行为。下图展示了一个真实场景下的异构网络：



传统的 Graph Embedding 方法比如 DeepWalk 的做法都是直接忽略图中边的类型以及节点的特征，将真实网络建模为 HON。而如果将边类型考虑进去建模为 MHEN，则会取得非常明显的效果。最后，如果还能够将节点的属性也考虑进去建模为 AMHEN，那么就充分利用到了原图的所有信息，这样便可以得到最好的效果。

除了异构性和多样性外，处理 AMHEN 也面临着多重挑战：

- 多路复用的边（Multiplex Edges）：每个节点对可能含有多种不同的类型边；
- 归纳学习（Inductive）：如何解决冷启动问题；
- 可扩展性（Scalability），如何拓展到大规模网络中。

为了解决这些问题，作者提出了一种一种通用的多路复用异构网络嵌入学习框架（General Attributed Multiplex HeTerogeneous Network Embedding，GATNE），并用于捕获节点丰富的属性信息和来自不同节点类型的多路复用拓扑结构。

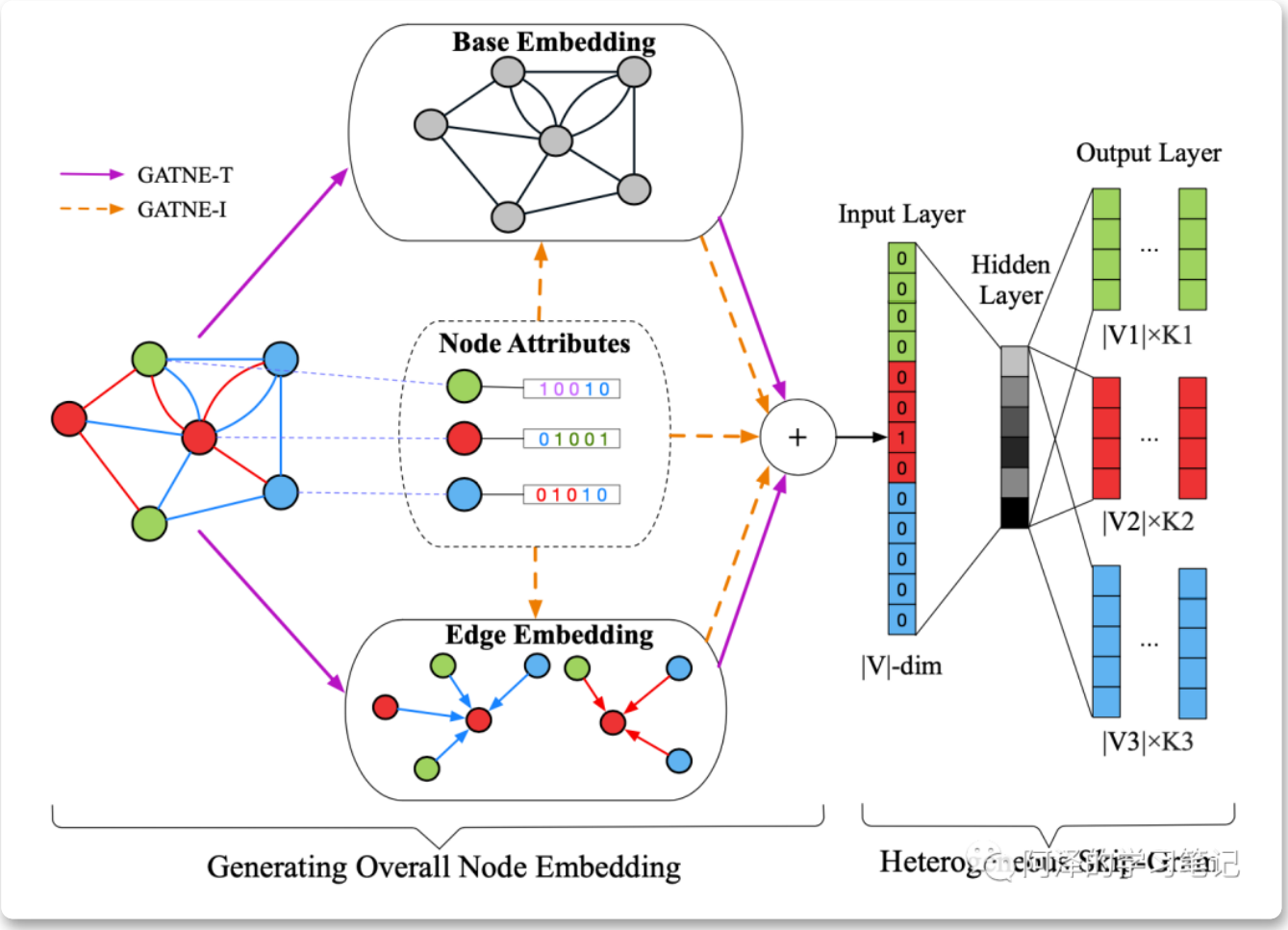
2.GATNE

本节介绍两种类型的 GATNE 框架，包括直推式学习范式的 GATNE-T 和归纳式学习范式的 GATNE-I。

2.1 GATNE-T

我们先从多路复用的异构网络图开始，并介绍 GATNE-T 模型。

GATNE-T 考虑 Base Embedding 和 Edge Embedding（这里的 Edge Embedding 并不是对边进行 Embedding，而是基于 edge type 进行聚合得到 Embedding），如下图紫色边所示。



节点的 Base Embedding 对所有类型的边共享。

第 k 层 v_i 节点的 edge type r 类型的 Edge Embedding 如下所示：

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

其中，$\mathcal{N}_{i,r}$ 为节点 v_i 基于 edge type r 的邻居；$u_{i,r}^{(0)}$ 是随机初始化的。

聚合函数可以为均值聚合、最大池化聚合等：

$$\mathbf{u}_{i,r}^{(k)} = \sigma\left(\hat{\mathbf{W}}^{(k)} \cdot \text{mean}\left(\left\{\mathbf{u}_{j,r}^{(k-1)}, \forall v_j \in \mathcal{N}_{i,r}\right\}\right)\right)\mathbf{u}_{i,r}^{(k)} = \max\left(\left\{\sigma\left(\hat{\mathbf{W}}_{\text{pool}}^{(k)}\mathbf{u}_{j,r}^{(k-1)} + \mathbf{b}\right.\right.\right.$$

使用 K 层的表示 $\mathbf{u}_{i,r}^{(K)}$ 作为边 $\mathbf{u}_{i,r}$ 的 Embedding，然后拼接节点 v_i 的所有 Edge Embedding：

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

然后利用 Self-Attention 来计算各边的权重：

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

其中，$\mathbf{w}_r$ 和 $\mathbf{W}_r$ 为可训练参数。

对于 edge type r 来说，节点 v_i 的 Embedding 为：

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

其中，$\mathbf{b}_i$ 为节点 v_i 的 base embedding，α_r 为超参用于控制 edge embedding 的重要程度，$\mathbf{M}_r$ 为可训练的转移矩阵。

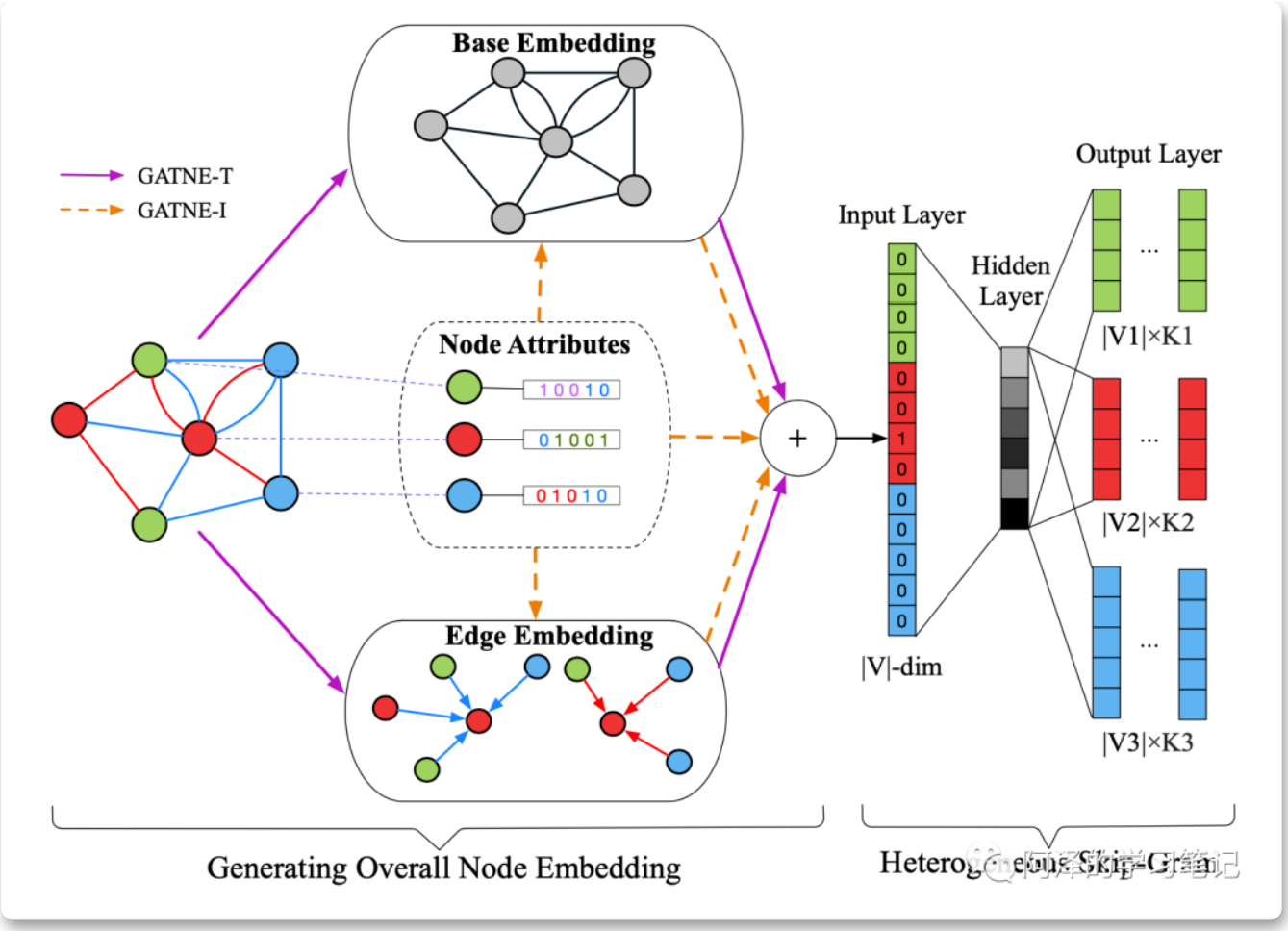
后面作者对比了 GATNE-T 和 MNE（Scalable Multiplex Network Embedding），并证明 GATNE-T 是 MNE 的一种泛化形式。

直推式模型 GATNE-T 在聚合的时候按照边的类型进行了分类，生成节点 v_i 对于 edge type r 的 Embedding。而在计算点 $v_{i,r}$ 时，使用了注意力机制为不同类型的边分配了不同的注意力。

但 GATNE-T 只是聚合了节点的邻居信息，没有应用到节点的属性信息。且由上面的公式可知，$u_{i,r}$ 都是通过聚合邻居得到的，训练的参数都是一个整体的矩阵。所以，GATNE-T 不能单独为新加入的节点生成 Embedding，也就是不能使用训练集训练好的参数用于生成（训练时不可见的）测试集的节点嵌入表示，必须重新训练。即 GATNE-T 只能进行直推式学习（transductive learning），不能进行归纳式学习（inductive learning）。

2.2 GATNE-I

为了解决 GATNE-T 的局限，作者提出了 GATNE-I 模型。该模型如下图橘色边所示：



在 GATNE-T 中，Base Embedding 和 Edge Embedding 都是随机初始化的，但是在 GATNE-I 中，这两个 Embedding 都是基于节点的特征，如上图中间部分，有两条橘色虚线从 Node Attributes 指向上下两方的 Embedding。

首先，定义 Base Embedding：

$$\mathbf{b}_i = \mathbf{h}_z(\mathbf{x}_i)$$

其中，$\mathbf{x}_i$ 为节点 v_i 的属性，$\mathbf{h}_z$ 为转换函数，$z = \phi(i)$ 映射节点 v_i 的边类型。

不同类型的节点其维度可能也不同，所以转换函数会有着不同的形式。

然后，定义 Edge Embedding：

$$\mathbf{u}_{i,r}^{(0)} = \mathbf{g}_{z,r}(\mathbf{x}_i)$$

其中，$g_{z,r}$ 也是一个转换函数，将属性特征转换为节点 v_i 关于 r 类型的 Edge Embedding。

于是，节点 v_i 在 edge type r 下的向量表示为：

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

其中，α_r, β_r 为系数，D_z 是针对属性的特征变换矩阵。

所以 GATNE-T 和 GATNE-I 的主要区别在于 Embedding 的初始化过程。

这边可能会有一些疑问：**「为什么初始化的方式不同会导致两种学习范式？」**

- 在 GATNE-T 中，$\mathbf{b}_i, \mathbf{u}_{i,r}^{(0)}$ 是基于网络结构的，为每个节点直接训练，所以无法处理训练中未出现过的节点；
- 而在 GATNE-I 中，训练的是两个转换函数 $\mathbf{h}_z, \mathbf{g}_{z,r}$，将原始特征 $\mathbf{x}_i$ 转换为 $\mathbf{b}_i, \mathbf{u}_{i,r}^{(0)}$，并非为每个节点直接训练，所以这便可以处理未出现的节点，只要这个节点有特征即可。

> 这边会有一个疑问，针对 GATNE-I 随机初始化原始特征 $\mathbf{x}_i$ 会怎么样。

2.3 Model Optimization

我们再来看一下模型的训练方式。

GATNEE 模型的训练方式是基于 meta-path 的随机游走和 heterogeneous skip-gram。

具体来说，我们先给一个预定的 meta-path scheme，如 user-item-user 等，然后给出随机游走的概率矩阵：

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

其中，$v_i \in \mathcal{V}_t$，$\mathcal{N}_{i,r}$ 为节点 v_i 基于 edgee type r 的邻居。

对于给定节点 v_i 和它的上下文 C，我么嗯的目标是最小化负对数似然函数：

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

在给定 v_i 时，v_j 的概率为：

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

其中，c_k 表示节点 v_k 的上下文 Embedding。

最后得到我们的目标函数为：

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

其中，$\sigma(x)$ 为 sigmoid 函数，L 为负采样的数量。

整体的算法流程为：

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

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

3.Experiments

简单看一下实验。

首先给出数据集：

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

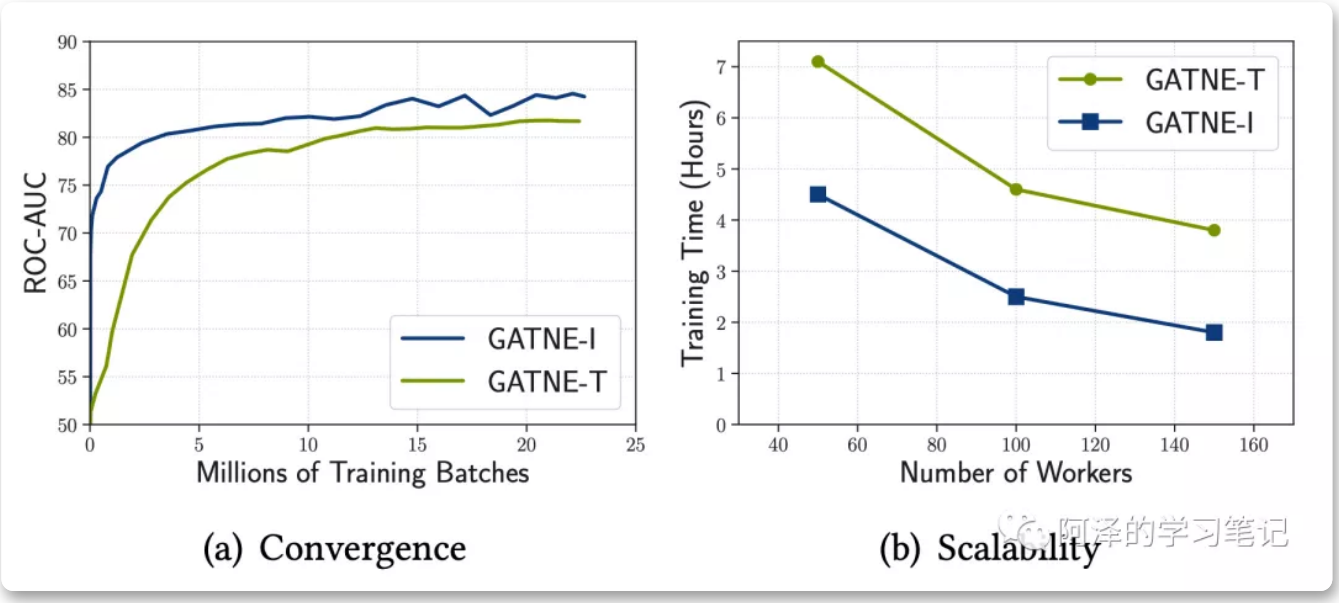
不同数据集下各模型的表现如下图所示，Amazon 的数据集节点的特征太少了，所以 GATNE-I 的效果略微逊色。

	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	**84.9**	63.71	67.53	52.48
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**

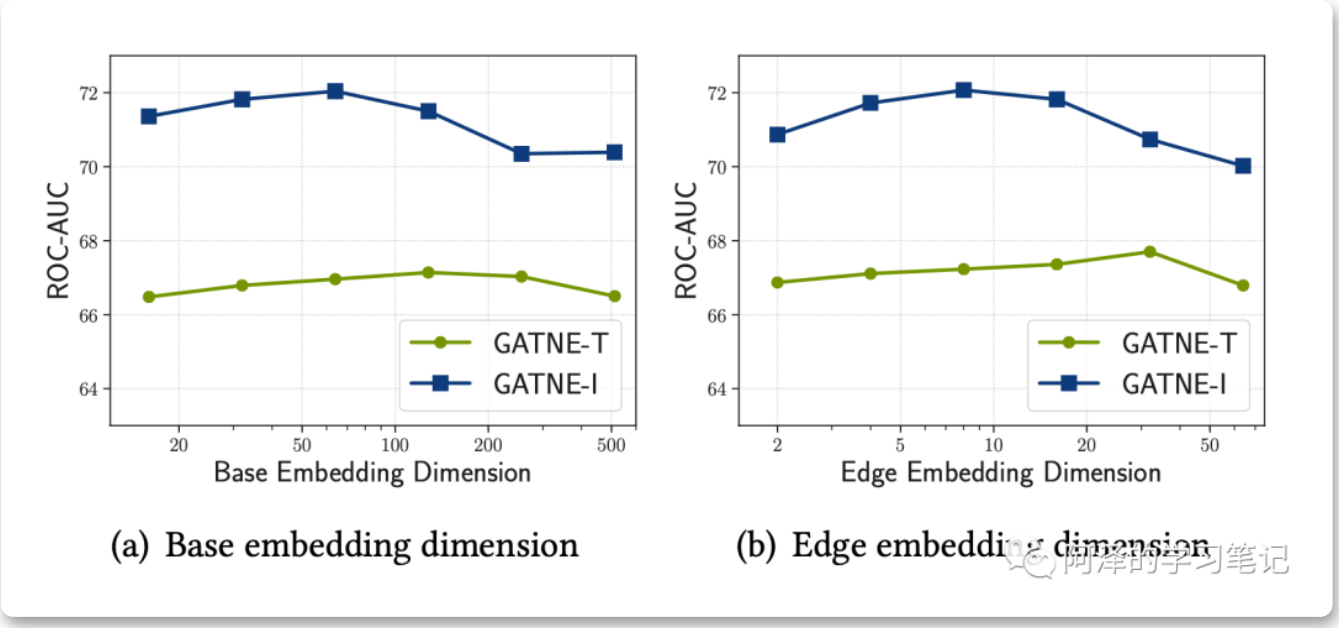
阿里巴巴数据集的表现如下图所示，阿里巴巴的数据集特征丰富，所以 GATNE-I 的效果很好。

	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**

收敛速度和可扩展性：



参数敏感性：



4.Conclusion

总结：本文首先基于网络拓扑类型和属性特征将图分为六大类，并重点关注目前研究较少的 MHEN 和 AMHEN 两种网络架构，分别提出了 GATNE-T 和 GATNE-I 两种模型。GATNE-T 在 MHEN 网络中建模并考虑 Base Embedding 和 Edge Embedding，Edge Embedding 利用注意力机制聚合邻居信息，并综合 Base Embedding 得到最终的 Node Embedding，GATNE-I 在 GATNE-T 的基础上考虑 Attribute Embedding 并弥补了 GATNE-T 无法泛化到未知节点的缺点。GATNE 模型在诸多工业界数据集中都取得不错的成绩，并在阿里推荐系统中成功落地应用。