

# 基于边采样的网络表示学习模型\*

陈丽<sup>1</sup>, 朱裴松<sup>1</sup>, 钱铁云<sup>1</sup>, 朱辉<sup>2</sup>, 周静<sup>2</sup>

<sup>1</sup>(武汉大学 计算机学院, 湖北 武汉 430072)

<sup>2</sup>(北京汇通金财信息科技有限公司, 北京 100094)

通讯作者: 钱铁云, E-mail: qty@whu.edu.cn



**摘要:** 近年来,以微博、微信、Facebook 为代表的社交网络不断发展,网络表示学习引起了学术界和工业界的广泛关注.传统的网络表示学习模型利用图矩阵表示的谱特性,由于其效率低下、效果不佳,难以应用到真实网络中.近几年,基于神经网络的表示学习方法因算法效率高、较好地保存了网络结构信息,逐渐成为网络表示学习的主流算法.网络中的节点因为不同类型的关系而相互连接,这些关系里隐藏了非常丰富的信息(如兴趣、家人),但所有现存方法都没有区分节点之间边的关系类型.提出一种能够编码这种关系信息的无监督网络表示学习模型 NEES (network embedding via edge sampling).首先,通过边采样得到能够反映边关系类型信息的边向量;其次,利用边向量为图中每个节点学习到一个低维表示.分别在几个真实网络数据上进行了多标签分类、边预测等任务,实验结果表明:在绝大多数情况下,该方法都表现最优.

**关键词:** 网络表示学习;图嵌入;深度学习;多关系类型;边采样

中图法分类号: TP18

中文引用格式: 陈丽,朱裴松,钱铁云,朱辉,周静.基于边采样的网络表示学习模型.软件学报,2018,29(3):756-771. <http://www.jos.org.cn/1000-9825/5435.htm>

英文引用格式: Chen L, Zhu PS, Qian TY, Zhu H, Zhou J. Edge sampling based network embedding model. Ruan Jian Xue Bao/ Journal of Software, 2018, 29(3): 756-771 (in Chinese). <http://www.jos.org.cn/1000-9825/5435.htm>

## Edge Sampling Based Network Embedding Model

CHEN Li<sup>1</sup>, ZHU Pei-Song<sup>1</sup>, QIAN Tie-Yun<sup>1</sup>, ZHU Hui<sup>2</sup>, ZHOU Jing<sup>2</sup>

<sup>1</sup>(School of Computer Science, Wuhan University, Wuhan 430072, China)

<sup>2</sup>(Beijing Huitong Jincai Information and Technology Company Limited, Beijing 100094, China)

**Abstract:** With the development of online social networks such as Weibo, WeChat and Facebook, network representation learning has drawn widespread research interests from academia and industry. Traditional network embedding models exploit the spectral properties of matrix representations of graphs, which suffer from both computation and performance bottlenecks when applied to real world networks. Recently, a lot of neural network based embedding models are presented in the literature. They are computationally efficient and preserve the network structure information well. The vertices in the network are connected to various types of relations, which convey rich information. However, such important information are neglected by all existing models. This paper proposes NEES, an unsupervised network embedding model to encode the relations. It first obtains the edge vectors by edge sampling to reflect the relation types of the edges. Then, it uses the edge vectors to learn a low dimension representation for each node in the graph. Extensive experiments are conducted on several social networks and one citation network. The results show that NEES model outperforms the state-of-the-art methods in multi-label classification and link prediction tasks. NEES is also scalable to large-scale networks in the real world.

\* 基金项目: 国家自然科学基金(61572376); 111 计划(B07037)

Foundation item: National Natural Science Foundation of China (61572376); 111 Project (B07037)

本文由基于图结构的大数据分析与管理技术专刊特约编辑林学民教授、杜小勇教授、李翠平教授推荐.

收稿时间: 2017-07-10; 修改时间: 2017-09-05; 采用时间: 2017-11-07; jos 在线出版时间: 2017-12-05

CNKI 网络优先出版: 2017-12-06 15:23:31, <http://kns.cnki.net/kcms/detail/11.2560.TP.20171206.1523.011.html>

**Key words:** network embedding; graph embedding; deep learning; multi relation type; edge sampling

网络结构是现实世界中的一种常见结构,比如社交网络、论文合作者网络、生物网络等.这些网络庞大而复杂,隐含了丰富的知识,许多研究工作都以网络为基础进行开展,如用户画像<sup>[1]</sup>、内容推荐<sup>[2]</sup>、异常检测<sup>[3]</sup>等.随着计算机技术迅猛发展,以微博、微信、Facebook 为代表的社交网络已进入亿级节点时代,这对大规模网络的相关研究又提出了更新、更大的挑战.

网络表示学习(network embedding)正是为了应对这种挑战而出现的,它是众多大规模网络研究领域的一个重要基础<sup>[4]</sup>.表示学习(representation learning)能够把人从繁琐的特征工程中解放出来,其目的就是通过无监督的方法,从原始数据中自动识别出有价值的信息进行保存,并编码到一个低维、稠密、连续的向量空间中<sup>[5]</sup>.网络表示学习指表示学习在网络研究领域的特称,其目的是为网络中每个节点自动学习到合适的表示.

传统构建网络表示学习模型的特征工程方法通常需要人工参与来提取网络的结构性质作为节点的特征<sup>[6,7]</sup>,此类方法复杂繁琐难以自动化,且通常会针对特定领域,不具有推广性.此外,一些经典的无监督网络(图)表示学习方法通常会利用输入数据矩阵的谱(如特征值和特征向量、奇异值和奇异向量),统称基于谱方法的网络表示学习模型<sup>[8]</sup>.从线性代数的角度来看,这些方法通常被视为降维技术,如多维尺度变换(multidimensional scaling,简称 MDS)<sup>[9]</sup>、等距映射(isometric mapping,简称 IsoMap)<sup>[10]</sup>、局部线性嵌入(locally linear embedding,简称 LLE)<sup>[11]</sup>、拉普拉斯特征映射(Laplacian eigenmaps,简称 LE)<sup>[12]</sup>等方法,这些方法在计算效率上有一定的局限性,因为它们首先要利用  $K$  最近邻算法( $k$ -nearest neighbor algorithm,简称  $K$ -NN)<sup>[13]</sup>等模型得到一个相似度矩阵(affinity matrix),通过特征向量求解的方式来学习网络节点的表示,该过程模型的时间复杂度通常都达到  $O(n^2)$ <sup>[8]</sup>,因此难以拓展到大规模网络之上.

近期,自然语言处理领域在表示学习方法方面取得了显著的成果,其中具有代表性的是词嵌入(word embedding)<sup>[14]</sup>,即,基于神经网络进行单词的特征学习.该研究认为:具有相似上下文的单词,其语义也应该相似.从而提出 CBOW 模型与 Skip-Gram 模型.这种通过无监督学习方法得到的词向量已在众多任务中取得了优异的表现.

受上述方法启发,研究者开始将词嵌入技术应用于网络或图节点的特征学习.最早进行此项工作的是 Perozzi<sup>[15]</sup>,他发现:文本语料中词语出现的次数和在网络中随机游走节点被访问的次数,两者都服从指数分布.从而认为,在语言模型中表现优异的 Skip-Gram 模型也可以移植到网络节点的表示学习中.以此提出了 DeepWalk 模型.该模型通过在图中进行随机游走获得一系列节点序列,然后将每个节点序列视为一个句子,作为 Skip-Gram 模型的输入,最终得到每个节点的低维表示.该模型训练速度快,学习到的向量取得了较好的效果,但其可解释性不够强,并且没有区分邻居的类别.

文献[16]提出的 Node2Vec 模型对 DeepWalk 的游走策略提出了改进措施,该模型有两个参数:返回参数  $p$  进出参数  $q$ ,分别用于控制在一次游走过程中重返上一个节点的可能性,以及游走方式是趋向于深度遍历图还是广度遍历图.但是参数  $p, q$  的设置需要提供额外的验证集,并且耗时巨大.同样,该模型也没有区分关系类别,将邻居进行同一对待.

文献[17]提出了一种可规模化的网络节点表示学习方法 LINE(large-scale information network embedding).该模型的可解释性强,使用一阶相似性(相邻节点之间的相似性)保存局部结构,使用二阶相似性(拥有共同邻居的节点之间的相似性)获取全局结构.但缺点在于:只是简单地利用两个相似度训练出的向量进行拼接,两者没有进行有机结合.

网络中的节点因为不同类型的关系相互连接,这些关系里隐藏了非常丰富的信息.例如,一个节点可能因为各种原因与其他节点相连接,有的连接表示兴趣关系,有的连接表示家人关系.但是,所有现存网络表示学习方法都只关注节点间的边是否存在,而不关注边之间是否存在区别.为了解决上述问题,本文提出了可以编码关系类型信息的无监督网络表示学习模型 NEES(network embedding via edge sampling).主要由两个阶段组成.

- 第 1 阶段目标是利用边采样优化一个目标函数,为图中每条边学习一个低维表示,若两条边的关系类

型相似,则它们的向量也会相似.因目的在于学习边的向量,因此本文将第 1 阶段也称为边向量模型;

- 第 2 阶段目标则是利用第 1 阶段训练得到的边向量改进现有的网络表示学习模型 DeepWalk,因其目的是为每个节点学习一个低维表示,所以本文将第 2 阶段称为节点向量模型.

NEES 模型主要对 DeepWalk 模型进行了两处改进:一方面,我们加强对网络结构的一阶相似性的利用,若两个节点之间有边相连,则两节点表示相似;另一方面,我们将通过第一阶段训练得到的边向量,提高节点序列中两两相连的边的相似度,以此提高模型对二阶相似性保存的准确性.

为了验证 NEES 模型的效果,我们在几个真实世界的网络数据上进行了多标签分类和边预测任务,对比目前最新的一些节点特征学习的方法,结果证明:在绝大多数情况下,我们的方法都表现最优.我们还探索了模型各要素的影响,包括对参数的敏感程度以及模型对网络规模增长的耗时变化.

本文主要贡献:

(1) 提出了一种无监督网络表示学习模型 NEES,它不仅能够利用邻居节点信息,也能够利用节点与邻居的关系信息;

(2) 为区分关系类型提供了一种新的方式,它无需标记数据,仅利用网络本身的结构信息,就能够使得关系类型相似的边学习到相似的向量;

(3) 在几个真实世界的网络上进行了多标签分类和边预测任务,实验结果证明,我们的方法在多个应用场景下都能取得优异的结果.

本文第 1 节将具体介绍网络节点表示学习的代表性模型 DeepWalk,并在此基础上详细说明本文提出的 NEES 模型的原理.第 2 节为实验,第 3 节对本文进行总结,并阐述进一步的研究方向.

## 1 基于边采样的网络表示学习模型 NEES

本文提出无监督的网络表示学习模型 NEES,它以 DeepWalk 模型为基础,沿用了 DeepWalk 模型保存网络结构的一阶相似性和二阶相似性的方式,但它加强了对一阶相似性的利用,提高了对二阶相似性保警惕存准确性.我们首先给出网络表示学习以及节点一阶相似性和二阶相似性的定义,然后介绍 DeepWalk 模型以及本文提出的无监督模型 NEES,最后对模型投入实际使用可能遇到的问题进行讨论和分析.

### 1.1 问题定义

本文研究的问题是如何构建一个合适的网络表示学习模型,它能将图中的数据映射到一个低维向量空间,每个节点能够用一个低维向量进行表示,并且这些向量之间的相对关系反映了节点之间的一阶相似性和二阶相似性.形式化定义如下.

**定义 1(网络表示学习).** 给定一个图  $G=(E,V)$ ,其中  $V=\{v_1,\dots,v_n\}$  表示节点集合, $E=\{e_{ij}\}$  表示边集合.网络表示学习构建的目标是学习一个映射函数  $f:v_i\rightarrow v_i\in R^d$ ,其中, $d\ll|V|$ ,向量  $v_i$  和  $v_j$  的关系能够反映出图中节点  $v_i$  和  $v_j$  的关系, $d$  是确定向量维度的参数.

在现实世界的网络结构数据中,相连的两个节点之间通常具有一定程度的相似性.例如:在社交网络中,互相关注的两个用户很有可能具有相同的兴趣爱好;在互联网中,相互链接的两个网页可能呈现的是相似主题的内容.我们把这种相似性称为节点的一阶相似性.

**定义 2(一阶相似性).** 给定一个图  $G=(E,V)$ ,对于任意的两个节点  $v_i,v_j$ ,若两个节点之间存在边,即  $e_{ij}\in E$ ,则称  $v_i,v_j$  之间具有一阶相似性.

然而在现实世界中,只有少部分连接数据能够被获取,大部分连接信息丢失,因而也丢失了大量节点之间的一阶相似性信息.因此,仅仅依靠一阶相似性很难推断网络结构信息.为了解决这个问题,基于一阶相似性的定义,我们可以得到假设:拥有共同邻居的节点之间具有一定程度的相似性.例如:在社交网络中,同时关注了某个明星的两个用户很有可能具有相同的兴趣爱好.在文献引用网络中,同时引用同一篇文章的两篇文章很可能具有相似的研究方向.我们把这种相似性称为节点的二阶相似性.

**定义 3(二阶相似性).** 给定一个图  $G=(E,V)$ ,对于任意的两个节点  $v_i,v_j$ ,若两个节点之间存在公共邻居,即存

在节点  $v_k$ ,使得  $e_{i,k} \in E, e_{j,k} \in E$ ,则称  $v_i, v_j$  之间具有二阶相似性.

### 1.2 DeepWalk模型

DeepWalk 模型是词表示学习在网络表示学习领域的一种应用.该模型通过在网络中随机游走生成节点序列,然后将节点序列视为词序列,应用 Skip-Gram 模型为每个节点生成分布式表示.与 Skip-Gram 模型一样,DeepWalk 的目标函数为最大化目标节点和其上下文节点在一个随机窗口内共现的概率.具体来说,设有一条随机游走序列  $s = \{v_1, v_2, \dots, v_l\}$ ,  $l$  为序列长度.设窗口大小为  $w$ ,则对于每一个目标节点  $v_i$ ,它的上下文节点定义为  $c_i = \{v_{i-w}, \dots, v_{i+w}\} \setminus v_i$ .则 DeepWalk 的目标函数为

$$L(S) = \sum_{s \in S} \left[ \frac{1}{l} \sum_{i=1}^l \sum_{v_j \in c_i} \log P(v_j | v_i) \right],$$

其中,  $S$  为随机游走序列的集合.  $P(v_j | v_i)$  由 softmax 函数计算得到:

$$P(v_j | v_i) = \frac{\exp(x_j \cdot x_i)}{\sum_{v \in V} \exp(x_v \cdot x_i)},$$

其中,  $x_i, x_j$  为节点  $v_i$  和  $v_j$  的向量表示.由于分母部分计算量过大,文献[15]中,通过层次 softmax 和负采样两种方法来加速计算.

### 1.3 边向量模型

NEES 模型总体分为两个阶段,第 1 阶段,边向量模型;第 2 阶段,节点向量模型(见第 1.4 节).边向量模型是利用边采样优化一个目标函数,为图中每条边学习一个低维表示,若两条边的关系类型相似,则它们的向量也会相似.它是 NEES 模型的难点所在,也是本文的核心部分.我们注意到,一个节点会因为不同的关系类型与其他节点聚集在一起,所以一个节点的邻居可以划分为不同的邻居聚簇,且不同邻居聚簇间的关系类型不同.也就是说,我们只需要训练一个模型,使得同一个邻居聚簇内边的相似度高于一聚簇外边的相似度即可.

影响聚簇算法效果的因素很多,且对每个节点的邻居进行聚簇,算法时间复杂度太大,所以我们需要将目标继续进行简化.这里,首先引入一个自我中心网络(ego-network)的概念,见定义 4.

**定义 4(自我中心网络).** 给定一个图  $G=(E, V)$ ,对于任意的一个节点  $v_i$ ,其自我中心网络为  $G'=(E', V')$ ,其中,  $V'$  为该节点和其所有邻居节点的集合,  $E'$  为  $V'$  中所有节点之间的边的集合.

每个节点都有其对应的自我中心网络,该网络包括该节点及其邻居,以及节点与邻居之间的边、邻居与邻居之间的边.如图 1(a)所示,即为节点  $a$  的自我中心网络.

假定节点  $a$  的邻居划分为  $k_1$  和  $k_2$  两个邻居聚簇,节点  $b$  和节点  $c$  都属于聚簇  $k_1$ .我们发现,出现在聚簇  $k_1$  中的边,大都也出现在节点  $b$  和节点  $c$  的自我中心网络中,如图 1(b)、图 1(c)所示.总体上看,聚簇越紧密,则越多该聚簇内的边同时出现在该聚簇内节点的自我中心网络内;反之,我们可以认为,若多条边同时出现在多个自我中心网络中,这多条边应当属于同一个聚簇.如此一来,我们不但可以避免显式调用聚簇算法对节点的邻居进行聚簇,而且可以利用网络结果自身的性质进行隐式聚簇.

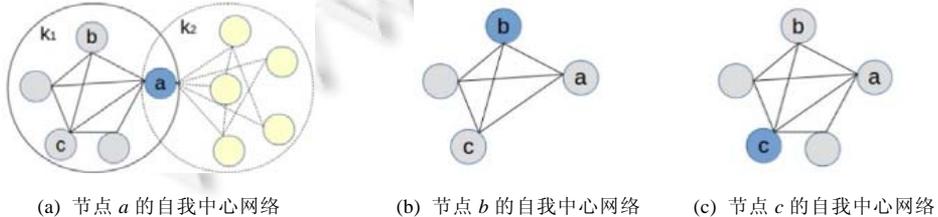


Fig.1 Ego-Networks of node  $a, b, c$

图 1 节点  $a, b, c$  的自我中心网络

为使同一自我中心网络的边向量相似度高于一不同自我中心网络的边向量相似度,定义如下目标函数:

$$\max \sum_{v \in V} \sum_{e \in E_v} \log P(v|e).$$

给定节点  $v, E_v$  是节点  $v$  的自我中心网络中边的集合,  $P(v|e)$  是当出现边  $e$  时, 网络为节点  $v$  的自我中心网络的概率. 为了达到让同一自我中心网络的边向量相似的目的, 我们将概率函数构造与 Skip-Gram 相似, 将其看做二分类问题, 使用逻辑回归作为分类方法. 为了加快训练速度, 我们使用负采样技术, 对于给定节点  $v$ , 其负样本集合为  $NEG(v)$ . 对于  $\forall u \in V$ , 首先定义如下指示函数:

$$I^v(u) = \begin{cases} 1, & u = v \\ 0, & u \neq v \end{cases}.$$

然后, 利用负采样技术概率函数构造如下:

$$P(v|e) = \prod_{u \in \{v\} \cup NEG^e(v)} P(u|e),$$

其中,

$$P(u|e) = \begin{cases} \sigma(e^T \theta^u), & I^v(u) = 1 \\ 1 - \sigma(e^T \theta^u), & I^v(u) = 0 \end{cases},$$

其中,  $\sigma$  为 sigmoid 函数;  $\theta^u$  是  $u$  的自我中心网络的向量, 是待定参数;  $e$  为边  $e$  的向量, 既是待定参数, 也是最终要输出的边向量, 它由边两端节点的向量进行位操作得到. 知识图谱表示学习模型 TransE 和 TransR 中边的表示使用减法操作, 即  $t-h=r$ . 因为知识图谱中的边都是有向边, 使用减法相对合理. 本文为了同时适应有向图和无向图, 使用的是求平均操作, 即: 如果边  $e$  两端的节点分别为  $v_i$  和  $v_j$ , 表示为  $e_{i,j}$ , 对应的边向量:

$$e_{i,j} = \frac{v_i + v_j}{2}.$$

最终的目标函数即为

$$\max \sum_{v \in V} \sum_{e_{i,j} \in E_v} \sum_{u \in \{v\} \cup NEG^{e_{i,j}}(v)} \{I^v(u) \cdot \log[\sigma(e_{i,j}^T \theta^u)] + [1 - I^v(u)] \cdot \log[1 - \sigma(e_{i,j}^T \theta^u)]\}.$$

为方便起见, 我们将大括号内的内容简记为  $L(v, e, u)$ , 即

$$L(v, e, u) = I^v(u) \cdot \log[\sigma(e_{i,j}^T \theta^u)] + [1 - I^v(u)] \cdot \log[1 - \sigma(e_{i,j}^T \theta^u)].$$

我们假设各参数之间独立, 采用梯度下降法对上式进行优化, 关键点在于要给出  $L(v, e, u)$  关于  $\theta^u$  的梯度和  $e_{i,j}$  的梯度. 首先, 我们考虑  $L(v, e, u)$  关于  $\theta^u$  的梯度:

$$\begin{aligned} \frac{\partial L(v, e_{i,j}, u)}{\partial \theta^u} &= \frac{\partial}{\partial \theta^u} \{I^v(u) \cdot \log[\sigma(e_{i,j}^T \theta^u)] + [1 - I^v(u)] \cdot \log[1 - \sigma(e_{i,j}^T \theta^u)]\} \\ &= I^v(u) [1 - \sigma(e_{i,j}^T \theta^u)] e_{i,j} - [1 - I^v(u)] \sigma(e_{i,j}^T \theta^u) e_{i,j} \\ &= [I^v(u) - \sigma(e_{i,j}^T \theta^u)] e_{i,j}. \end{aligned}$$

$\theta^u$  的更新公式为

$$\theta^u + = \eta [I^v(u) - \sigma(e_{i,j}^T \theta^u)] e_{i,j},$$

其中,  $\eta$  为学习率. 然后, 我们再考虑  $L(v, e, u)$  关于  $e_{i,j}$  的梯度. 由于  $e_{i,j}$  和  $\theta^u$  在  $L(v, e, u)$  具有对称性, 易得:

$$\frac{\partial L(v, e_{i,j}, u)}{\partial e_{i,j}} = [I^v(u) - \sigma(e_{i,j}^T \theta^u)] \theta^u.$$

根据连续求导法则以及  $v_i$  和  $v_j$  在  $e_{i,j}$  中具有对称性, 所以,

$$\frac{\partial L(v, e_{i,j}, u)}{\partial v_i} = \frac{1}{2} [I^v(u) - \sigma(e_{i,j}^T \theta^u)] \theta^u.$$

$v_i$  的更新公式为

$$v_i + = \frac{\eta}{2} \sum_{u \in \{v\} \cup NEG^{e_{i,j}}(v)} [I^v(u) - \sigma(e_{i,j}^T \theta^u)] \theta^u.$$

$v_j$  与  $v_i$  更新公式相同.

将一个节点自我中心网络内的边作为输入, 利用随机梯度下降法优化, 则每次输入后, 会使得某个节点自我

中心网络内边的向量表示变得更相似;输入多个节点的自我中心网络后,会出现如下情况.

- 同一聚簇内,边之间的相似度高于一自我中心网络内边之间的相似度.因为根据梯度下降法易知:边表示更新次数越多,就越趋近于满足目标函数.而同一聚簇的边,如聚簇  $k_1$  中的边,由于出现在节点  $a,b,c$  等多个不同的自我中心网络内,所以向模型输入节点  $a,b,c$  的自我中心网络时,聚簇  $k_1$  内的边的向量都会被更新,其相似性会不断加强;
- 同一聚簇内,边与在同一自我中心网络内、但在同一邻居聚簇的边相似性会减弱.因为不在该聚簇内的边很少会出现在该聚簇内节点的自我中心网络内,如聚簇  $k_2$  中的边不出现在节点  $b,c$  的自我中心网络内,所以向模型输入节点  $b,c$  的自我中心网络时,通过负采样可能会使得处于  $b,c$  的自我中心网络且属于聚簇  $k_1$  的边,与不处于  $b,c$  的自我中心网络且属于聚簇  $k_2$  的边变得不相似.

两个方向同时增强,即使没有直接对节点的自我中心网络进行聚类,但是可间接使得同一聚簇的边的相似度高于一聚簇外的边的相似度.

最后,我们将边向量模型总结如下.

**算法 1. NEES——边向量模型.**

输入:图  $G=(V,E)$ , 维度  $d1$ , 负采样个数  $n1$ ;

输出:边向量矩阵  $\Phi \in R^{|E| \times d1}$ .

1. 基于均匀分布初始化边向量矩阵  $\Phi$
2. For each  $v \in V$  do
3. For each  $e_{i,j} \in E_v$  do
4.  $e_{i,j} = (v_i + v_j) / 2$
5. 对节点  $v$  进行负采样,得到负样本集合  $NEG^{e_{i,j}}(v), |NEG^{e_{i,j}}(v)| = n1$
6. For each  $u \in \{v\} \cup NEG^{e_{i,j}}(v)$  do
7.  $\theta^u += \eta [I^v(u) - \sigma(e_{i,j}^T \theta^u)] e_{i,j}$
8. End for
9.  $v += \frac{\eta}{2} \sum_{u \in \{v\} \cup NEG^{e_{i,j}}(v)} [I^v(u) - \sigma(e_{i,j}^T \theta^u)] \theta^u$
10. End For
11. End For

#### 1.4 节点向量模型

NEES 模型的第 2 阶段也称为节点向量模型,其目标是利用第 1 阶段训练得到的边向量改进现有的网络表示学习模型 DeepWalk.像文献[15,16]一样,首先通过在图中进行随机游走,得到一系列的节点序列.但我们采取了有偏向的随机游走,即:通过第 1 阶段学习得到的边向量,增加游走前后两条边向量的相似性,如此可以增加模型对网络结构二阶相似性保存的准确性.然后,将节点序列作为 Skip-Gram 模型的输入,此处我们对 Skip-Gram 模型也进行了改进,原模型只是间接地保存了部分一阶相似性,我们在这一方面进行了改进,增强了直接相连的节点之间的相似性.

现有方法通常将二阶相似性定义为具有相似邻居的节点相似.NEES 模型则将其定义为:具有共同邻居且与共同邻居关系类型相似的节点相似.具体实现就是:通过第 1 阶段学习得到的边向量,增加游走前后两条边的向量的相似性来完成的.例如,如果我们得到序列  $\langle a,b,c,d,e \rangle$  和  $\langle a,b,f,d,e \rangle$  (如图 2 中节点序列(1)、序列(2)所示),使得节点  $c$  和  $f$  具有相似上下文,从而学习到相似的代表.那么根据 NEES 模型序列生成规则——序列中两两相连的边需具有较高的相似性,我们可以推断出若要得到上述节点序列,边向量  $bc$  要与  $ab$  相似,边向量  $bf$  也要与  $ab$  相似,即边向量  $bc$  与  $bf$  是相似的;同理,边向量  $cd$  与  $fd$  也是相似的.也就是说,NEES 模型中的节点序列,若两个序列节点相似,则两个序列中各边也是相似的.又因为 NEES 模型的边向量模型是令具有相似关系类型的边的

向量相似,所以若两个序列节点相似,则两个序列中各边的关系类型是相似的.反之,若与共同邻居的关系不同,如图2中节点序列(2)、序列(3)所示:随着序列延伸,节点*f*和*g*的上下文相似度将降低,也就是说,节点*f*和*g*最后学习到的向量表示的相似度也会降低.

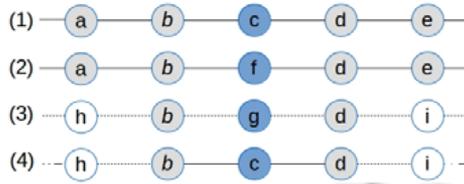


Fig.2 An example of the influence of relation type information on the node sequences (same type lines mean similarity relation type)

图2 关系类型信息对节点序列构成的影响示例(相同类型的线表示相似的关系类型)

NEES 模型的游走方式改善了生成的节点序列的准确性,并且通过提高相连边的相似度,可以使得整个序列中的节点都存在某种联系.而按照以前的随机游走方式,节点序列中只有前后相邻的两个节点具有较强的联系,且关系类型不确定.在这种情况下,因具有相似上下文而学到的相似表示则不具有较高的可信度.比如,若按以前的随机游走方式,我们可能会得到如图2中所示的(2),(3),(4)节点序列,*c*与*f*可能会具有相似的左右邻居*b*和*d*,但由于不确定关系类型,前后延伸的节点中很难有机会同时再出现*a*和*e*,导致*c*和*f*的上下文相似度会极大地降低;相反,若*c*和*g*与共同邻居*b*和*d*的关系类型不相似,即使有相似的上下文,*c*与*g*相似的结论也是不可信的.

具体实现方面,在第1阶段的边向量模型训练后,我们得到一张带有边向量的图,边向量保存了图的关系类型信息.此时,我们再沿用DeepWalk模型,在图上进行有偏向的随机游走,得到一系列节点序列:对给定的游走起点*w*<sub>0</sub>,我们将得到一条长为*l*的节点序列,*w*<sub>*i*</sub>表示序列中第*i*个节点,若当前节点为*w*<sub>0</sub>,则从其邻居中随机选择一个节点作为*w*<sub>1</sub>;若当前节点为*w*<sub>*k*</sub>(*k*≥1),则*w*<sub>*k*+1</sub>的选择遵循如下概率分布:

$$P(w_{k+1} = x | w_k = v, w_{k-1} = t) = \begin{cases} \frac{\pi(t, v, x)}{Z}, & e_{v,x} \in E \\ 0, & \text{otherwise} \end{cases}$$

其中,*Z*为归一化常数; $\pi(t, v, x)$ 为当从节点*t*游走到节点*v*后,再从节点*v*游走到节点*x*的转移概率:

$$\pi(t, v, x) = \begin{cases} \mu, & \text{if } x = t \\ \text{similarity}(e_{t,v}, e_{v,x}), & \text{if } x \neq t \text{ and } e_{v,x} \in E. \\ 0, & \text{otherwise} \end{cases}$$

此处, $\mu$ 是返回参数.由于它并不是本文的重点,我们将其统一设置为0.5.另外,*similarity*的计算方式,我们采用的是余弦相似度,即

$$\text{similarity}(e_{t,v}, e_{v,x}) = \frac{e_{t,v} \cdot e_{v,x}}{\|e_{t,v}\| \cdot \|e_{v,x}\|}$$

其中,*e*<sub>*t,v*</sub>和*e*<sub>*v,x*</sub>分别是边*e*<sub>*t,v*</sub>和边*e*<sub>*v,x*</sub>对应的向量,它们是由第1阶段的边向量训练模型学习得到的.

依次将图中每个节点作为序列的起始节点*w*<sub>0</sub>,根据邻居节点选择概率分布对邻居进行采样.采样时,可利用别名采样法(alias sampling algorithm)<sup>[18]</sup>进行,它能让采样过程在*O*(1)的时间复杂度内完成.对每个*w*<sub>0</sub>,我们在图上进行有偏向的随机游走,得到长度为*l*的序列.上述操作重复*r*次后,我们将上述游走得到的节点序列都输入到改进过的Skip-Gram模型,训练得到最终的节点向量.如前所述,我们增强了对一阶相似性的利用,Skip-Gram模型原本的目标函数为 $\prod_{\tilde{w} \in C(w)} p(w | \tilde{w})$ ,能达到让具有相似上下文的节点学习到相似的向量,现在我们更改为

$$\prod_{e_{i,j} \in E} p(v_i, v_j) \prod_{\tilde{w} \in C(w)} p(w | \tilde{w}),$$

其中,  $p(v_i, v_j)$  用于保存一阶相似性, 定义为

$$p(v_i, v_j) = \sigma(v_i^T v_j),$$

其中,  $v_i$  和  $v_j$  分别是节点  $v_i$  和  $v_j$  作为上下文节点时的向量表示. NEES 模型通过控制邻居的选择策略, 可以提高得到的节点序列中相连的边的相似度. 此时, 若节点  $c$  和  $f$  具备相似的上下文, 如图 2 中的序列 (1), (2) 所示, 它们与共同邻居的节点  $b$ 、节点  $d$  的关系也是类似的. 而序列输入改进的 Skip-Gram 模型后, 会让具有相似上下文的节点  $c$  和节点  $f$  学习到的向量相似. 也就是说, 我们达到了令具有共同邻居、且与共同邻居关系类型相似的节点学习的向量相似的目的. 该节点向量训练模型还学习了 LINE 模型的优点, 对一阶相似性进行了更为完整的保存.

节点向量模型总结如下.

**算法 2.** NEES 节点向量模型.

输入: 图  $G=(V, E)$ , 边向量矩阵  $\Phi \in \mathbb{R}^{|E| \times d_1}$ , 返回参数  $\mu$ , 游走次数  $r$ , 游走长度  $l$ , 窗口  $w$ , 维度  $d_2$ , 负采样个数  $n_2$ ;

输出: 节点向量矩阵  $\Psi \in \mathbb{R}^{|V| \times d_2}$ .

1. 初始化节点序列集合  $W$  为空
2. 计算转移概率  $\pi(t, v, x)$
3. For  $i=1$  to  $r$  do
4.  $V = \text{Shuffle}(V)$
5. For each  $v \in V$  do
6.  $s_v = \text{BiasedRandomWalk}(G, \pi, v, l)$
7. 将节点序列  $s_v$  加入集合  $W$
8. End for
9. End for
10.  $\text{ImprovedSkipGram}(G, W, \Psi, w, d_2, n_2)$

$\text{BiasedRandomWalk}(G, \pi, v, l)$

1. 初始化节点序列  $s_v = [v]$
2. For  $i = 1$  to  $l$  do
3. 当前节点  $cur = s_v = [v]$
4. 获取当前节点  $cur$  的邻居节点集合  $N_{cur} = \text{GetNeighbours}(G, cur)$
5. 依概率选择  $cur$  的邻居节点  $s = \text{AliasSample}(N_{cur}, \pi)$
6. 将  $s$  加入节点序列  $s_v[i+1] = s$
7. End for
8. Return  $s_v$

$\text{ImprovedSkipGram}(G, W, \Psi, w, d_2, n_2)$

1. 基于均匀分布初始化节点向量矩阵  $\Psi \in \mathbb{R}^{|V| \times d_2}$
2. For each  $e_{i,j} \in E$  do
3. For each  $s \in W$
4. For each  $v \in s$  do
5. 节点  $v$  的上下文  $C(v)$  为它在序列中前后个  $w$  节点
6. For each  $\tilde{v} \in C(v)$  do
7. 对节点  $v$  进行负采样, 得到负样本集合  $NEG^{\tilde{v}}(v), |NEG^{\tilde{v}}(v)| = n_2$
8. For each  $u \in \{v\} \cup NEG^{\tilde{v}}(v)$  do
9.  $\theta^u += \eta [I^v(u) - \sigma(\tilde{v}^T \theta^u)] \tilde{v}$
10. End for
11.  $\tilde{v}^+ = \eta \sum_{u \in \{v\} \cup NEG^{\tilde{v}}(v)} [I^v(u) - \sigma(\tilde{v}^T \theta^u)] \theta^u$

12. End For

13. End For

14. End for

总的来说,NEES 模型补充了 DeepWalk 模型对一阶相似性利用的不足,有机结合了两阶段的相似性,并且更精确地保存了图的二阶相似性.

### 1.5 复杂度分析

考量网络节点学习能否投入实际使用的另一个问题在于时空复杂度,本模型的空间复杂度主要受限于要同时保存所有节点的直接邻居,即空间复杂度为  $O(|E|)$ .时间复杂度可分为 3 个部分讨论.

- 第 1 阶段,边向量训练模型的时间复杂度为  $O(vkndi)$ ,其中, $v$  为图中节点数, $k$  为节点的平均度, $n$  为负采样的采样数, $d$  为向量的维度, $i$  为迭代次数.通常, $k$  对每个网络都有一个最大值,而  $n,d,i$  都是通过参数设定的常数,因此第 1 阶段模型的时间复杂度可以看做是与节点数  $v$  线性相关;
- 第 2 阶段包括随机游走和 Skip-Gram 模型训练,其中,随机游走的时间复杂度为  $O(vkdr_l)$ ,其中, $r$  是每个节点为起点游走的次数, $l$  是游走序列的长度,它们都是参数设定的常数,即,游走阶段也是与节点数  $v$  线性相关的;至于 Skip-Gram 模型,它的时间复杂度为  $O(swndi)$ ,其中, $s$  为输入文档中节点个数, $w$  为上下文窗口的大小,可通过参数设定.

Skip-Gram 模型的时间复杂度可以看做与输入的节点数  $s$  线性相关,且 3 个部分内部都可以并行处理,因此,NEES 模型可以作用在大规模的数据集上,我们将在实验部分对算法的时间消耗做出具体的呈现.

## 2 实验

本节对 NEES 模型进行量化分析实验.为了充分说明 NEES 模型的有效性,我们对其在边预测、多标签分类两个任务上的效果进行了实验.为了验证模型的鲁棒性和高效性,分别从模型对参数的敏感程度,以及随网络规模的时间消耗增长程度两个方面进行实验.

### 2.1 实验设置

NEES 模型训练出的向量不针对具体任务,只是单纯地保存图中节点和边之间的相对关系,可灵活地运用到各类任务中,比如边预测、节点分类等.对每个任务,我们都将 NEES 模型与下述模型进行对比.

- (1) DeepWalk<sup>[15]</sup>:该方法通过在图中进行随机游走得到节点序列,将序列输入 Skip-Gram 模型,为每个节点学习到一个  $d$  维向量表示.它可以看做 NEES 模型的一种特殊形式,即,所有边的相似程度都为 1;
- (2) LINE<sup>[17]</sup>:该方法分别利用一阶相似性和二阶相似性保存了网络的局部结构和全局结构,然后简单拼接两种方式学习到的表示,得到最终  $d$  维向量;
- (3) Node2Vec<sup>[16]</sup>:该方法是基于 DeepWalk 模型的,但是它通过两个参数控制了游走时邻居选择的策略,然后,同样将序列输入 Skip-Gram 模型为节点学习到  $d$  维向量表示.

文献[16]中提到,由于 DeepWalk 模型在优化阶段使用了基于层次 Softmax<sup>[19]</sup>的 Skip-Gram 模型,相比于负采样技术,它效果较好但较为低效,因此将 DeepWalk 中的 Skip-Gram 模型也转为基于负采样技术.本文与文献[16]做了相同处理,并保持其他条件一致.我们的参数默认值设置大部分都与文献[16]一致:负采样参数  $n_1$  和  $n_2$  均设为 5,向量维度  $d_1$  和  $d_2$  均设为 128,每个节点游走次数  $r$  为 10,序列长度  $l$  为 80,窗口大小  $w$  为 10,迭代次数为 1.至于 Node2Vec 模型的  $p,q$  参数,我们也是利用 10%的标记数据作为验证集,在  $p,q \in \{0.25,0.50,1,2,4\}$  中选取一对.对于 LINE,初始学习率为 0.025,总采样参数设为 100 亿.

### 2.2 评价指标

本文的实验部分主要进行两类任务:边预测和多标签分类任务.

- 对于多标签分类任务,像其他许多工作一样<sup>[3,20,21]</sup>,我们采用宏平均(Macro-F1)和微平均(Micro-F1)作为评价指标;

- 对于边预测任务,我们从网络中随机隐藏了一部分边作为测试样例,将剩余的图用于训练各个网络表示学习模型。

得到各节点向量后,计算两两节点间的余弦相似度,按相似度倒序排列,将训练数据中出现的边排除后,前  $k$  条边即为预测结果,是模型认为可能存在的边.与文献[22]类似,该任务使用前  $k$  精确率( $Precision@k$ )和平均精确率 MAP(mean average precision)作为评价指标.具体定义为

$$Precision@k = \frac{|\{e_{i,j} | v_i, v_j \in V, index(e_{i,j}) < k, \Delta_{i,j} = 1\}|}{k},$$

其中,  $V$  是图的节点集合,  $E'$  是原图隐藏的边集合,  $e_{i,j}$  表示节点  $v_i$  和  $v_j$  的边(该边不一定实际存在),  $index(e_{i,j})$  表示边  $e_{i,j}$  在模型预测结果中的排序序号,  $\Delta_{i,j}=1$  表示边  $e_{i,j}$  存在于边集  $E'$  中.

相比  $Precision@k$ , 平均精确率更注重预测结果中排列靠前的样例,它的计算公式如下:

$$AP = \frac{\sum_{i=1}^{|E'|} Precision@i \cdot \Delta_i}{|E'|},$$

$$MAP = \frac{\sum_{j=1}^Q AP(j)}{Q}.$$

其中,  $\Delta_i$  是指示函数,当第  $i$  个预测结果命中时,其值为 1,否则为 0;  $Q$  是查询次数.

### 2.3 多标签分类任务

多标签分类任务是指每个节点会被标记为一个或多个标签,它是节点特征学习效果衡量的一个常见的任务.本文也使用它来衡量 NEES 模型的学习效果.我们选择了 3 个社交网络数据集进行多标签分类任务,数据集统计特性见表 1.

Table 1 Statistical information of data sets

表 1 数据集的统计特性

数据集	$ V $	$ E $	平均度	标签数
BLOGCATALOG	10 312	333 983	64.87	39
FLICKR	80 513	5 899 882	146.56	195
YOUTUBE	1 138 499	2 945 443	5.25	47
YOUTUBE-small	22 693	96 361	8.49	47

训练时,首先随机选择一部分数据作为训练数据,剩下的作为测试数据.对于 BLOGCATALOG 数据集,我们随机选择了 10%~90% 作为训练数据;对于 FLICKR 数据集和 YOUTUBE 数据集,我们随机选择了 1%~10% 的数据作为训练数据.与文献[22]的处理方式一样,我们移除了 YOUTUBE 数据集中无标签的节点和孤立节点,得到网络 YOUTUBE-small.实验重复 5 次,并报告 5 次实验的  $Macro-F1$  均值和  $Micro-F1$  均值,结果见表 2(\*\*和\*分别表示 NEES 模型在置信度为 0.99 和 0.95 情况下显著优于该方法).

从表 2 中可以看出:NEES 模型除了在 BLOGCATALOG 数据集的 10% 和 20% 上的效果略低于其他模型,其余设置上均好于其他模型.尤其是在 BLOGCATALOG 数据集上,当训练数据比例为 90% 时,  $Macro-F1$  均值已经超过表现最好的 Node2Vec 模型 18%.这证明了我们的模型学习到的节点向量在多标签分类任务上.从总体上来看,Node2Vec 的效果要优于 DeepWalk 模型,只在 YOUTUBE 数据集上没有表现出优势,反而  $Micro-F1$  均值略低于 DeepWalk 模型.原因可能是 YOUTUBE 数据集的网络相对稀疏,邻居节点采样随机性减少,所以控制游走策略并不能带来明显的改进.相反,LINE 模型在最稀疏的 YOUTUBE 网络上表现优秀,反而在最稠密的 FLICKR 数据集上只与 Node2Vec 模型持平.这是因为 LINE 模型对一阶相似性保存得较好.NEES 模型因为不仅对游走策略进行了控制,也对一阶相似性的利用进行了加强,所以在各类数据集都取得了较好的表现.

**Table 2** *Macro-F1* scores and *Micro-F1* scores on BLOGCATALOG, FLICKR and YOUTUBE  
 表 2 BLOGCATALOG, FLICKR 和 YOUTUBE 数据集上的 *Macro-F1* 均值和 *Micro-F1* 均值

(a) BLOGCATALOG	Algorithm	NEES	Node2Vec	LINE	DeepWalk
<i>Macro-F1</i>	10%	<b>23.62</b>	22.13**	20.29**	20.06**
	20%	<b>26.93</b>	25.4*	22.94**	23.11**
	30%	<b>29.7</b>	26.72**	24.66**	24.04**
	40%	<b>31.62</b>	28.34**	25.59**	25.61**
	50%	<b>32.84</b>	28.45**	26.71**	25.77**
	60%	<b>34.48</b>	29.07**	27.58**	26.45**
	70%	<b>35.35</b>	30.49**	28.76**	27.29**
	80%	<b>36.68</b>	30.33**	29.07**	27.56**
	90%	<b>37.64</b>	30.81**	29.99**	27.85**
<i>Micro-F1</i>	10%	34.91	<b>36.38</b>	35.57	35.42
	20%	37.81	<b>38.96</b>	38.58	37.56
	30%	<b>40.01</b>	39.96	39.75	38.54**
	40%	<b>41.49</b>	41.12	40.93	39.22**
	50%	<b>42.61</b>	41.26**	41.51*	39.68**
	60%	<b>43.83</b>	41.83**	42.26**	40.37**
	70%	<b>44.59</b>	42.44**	43.09**	40.92**
	80%	<b>45.29</b>	42.58**	43.41**	40.98**
	90%	<b>46.08</b>	42.89**	44.05*	41.31**
(b) FLICKR	Algorithm	NEES	Node2Vec	LINE	DeepWalk
<i>Macro-F1</i>	1%	<b>13.82</b>	12.04**	9.57**	10.72**
	2%	<b>16.66</b>	14.36**	10.81**	12.65**
	3%	<b>18.47</b>	15.75**	11.96**	13.93**
	4%	<b>19.53</b>	16.62**	12.64**	14.66**
	5%	<b>20.48</b>	17.35**	13.42**	15.38**
	6%	<b>21.39</b>	18.17**	14.22**	16.02**
	7%	<b>22.09</b>	18.66**	14.87**	16.55**
	8%	<b>22.74</b>	19.13**	15.42**	16.9**
	9%	<b>23.13</b>	19.45**	15.95**	17.21**
	10%	<b>23.68</b>	19.78**	16.43**	17.51**
<i>Micro-F1</i>	1%	<b>33.35</b>	31.81**	32.51**	32.38**
	2%	<b>35.16</b>	33.71**	34.08**	33.61*
	3%	<b>36.25</b>	34.8**	35.02**	34.35*
	4%	<b>36.87</b>	35.53**	35.62**	34.76
	5%	<b>37.41</b>	36.18**	36.15**	35.12*
	6%	<b>37.84</b>	36.71**	36.6**	35.39*
	7%	<b>38.21</b>	37.2**	37.03**	35.64**
	8%	<b>38.48</b>	37.56*	37.27**	35.83**
	9%	<b>38.73</b>	37.93	37.6**	35.99**
	10%	<b>38.9</b>	38.2	37.89**	36.11**
(c) YOUTUBE	Algorithm	NEES	Node2Vec	LINE	DeepWalk
<i>Macro-F1</i>	1%	<b>33.92</b>	33.3*	28.35**	33.44*
	2%	<b>36.19</b>	35.32**	31.97**	35.58**
	3%	<b>37.28</b>	36.7*	34.37**	36.72*
	4%	<b>38.15</b>	37.5*	35.97**	37.49**
	5%	<b>38.85</b>	38.27*	37.19**	38.18**
	6%	<b>39.31</b>	38.77**	38**	38.65**
	7%	<b>39.74</b>	39.26*	38.62**	39.17**
	8%	<b>40.02</b>	39.68	39.1**	39.48**
	9%	<b>40.38</b>	39.88*	39.8**	39.77**
	1%	<b>33.92</b>	33.3*	28.35**	33.44*
<i>Micro-F1</i>	1%	<b>43.04</b>	38.75**	37.94**	39.03**
	2%	<b>44.66</b>	40.32**	40.8**	40.57**
	3%	<b>45.66</b>	41.4**	42.69**	41.61**
	4%	<b>46.3</b>	42.08**	43.93**	42.34**
	5%	<b>46.81</b>	42.92**	44.85**	43.03**
	6%	<b>47.19</b>	43.53**	45.43**	43.55**
	7%	<b>47.45</b>	43.95**	45.89**	44.03**
	8%	<b>47.59</b>	44.45**	46.15**	44.37**
	9%	<b>47.77</b>	44.73**	46.59**	44.72**
	10%	<b>47.97</b>	45.13**	46.96**	45.2**

## 2.4 边预测任务

边预测任务是衡量网络节点表示学习效果的另一个常见任务.对于该任务,我们使用 arXiv GR-QC 数据集<sup>[23]</sup>进行测试,这是一个论文合作者网络,此网络的论文限于 arXiv 上广义相对论和量子宇宙学领域.网络中,每个节点代表一个作者,若两个作者合作撰写了其中一篇论文,则两个节点间存在一条无向边.该网络共有 5 242 个节点,14 490 条边,平均度为 5.53.

为进行边预测任务,我们从网络中随机隐藏了一部分边作为测试样例,将剩余的图用于训练各网络表示学习模型.训练后,得到各节点向量,再计算两两节点间的余弦相似度.相似度较大的两个节点之间,我们认为可能存在一条边.

我们进行了两组实验:一方面评估方法对边预测任务的整体效果,另一方面评估模型在边预测任务上受图稀疏性的影响大小.

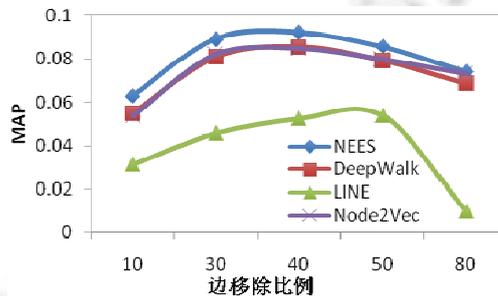
对于第 1 个实验,我们抽取了图中 15% 的边(大约 2 000 条边),使用  $Precision@k$  作为评判标准, $k$  值从 2 依次递增到 1 000,结果见表 3.从表中可以看出,NEES 模型在大多数情况下都表现略优于其他模型.这说明 NEES 模型在边预测任务上也能发挥较好的作用.

**Table 3**  $precision@k$  values of arXiv GR-QC dataset on link prediction task

**表 3** 边预测任务中,arXiv GR-QC 数据集上的  $precision@k$  值

	$P@2$	$P@10$	$P@100$	$P@200$	$P@300$	$P@500$	$P@800$	$P@1000$
NEES	1	0.7	0.42	0.34	0.37	0.33	0.32	0.3
DeepWalk	1	0.4	0.27	0.31	0.3	0.26	0.24	0.25
LINE	1	0.4	0.22	0.17	0.15	0.19	0.21	0.21
Node2Vec	0.5	0.5	0.4	0.36	0.3	0.26	0.25	0.24

对于第 2 个实验,我们改变了从图中抽取的边的比例,并使用  $MAP$  作为度量值,相较于  $precision@k$ ,它更关心排在前列的检索结果.实验结果如图 3 所示.结果表明,NEES 模型的结果始终优于其他 3 种模型.由于 NEES 模型与 DeepWalk,Node2Vec 模型相似,因此变化趋势也非常相似.而模型 LINE 在本数据集上的边预测效果表现较差,原因可能是 LINE 更依赖于二阶相似性,当我们直接将边移除,尤其是当移除边比例高达 80% 时,对二阶相似性的破坏更严重,所以 LINE 的效果出现较大的下降.除此之外,我们发现 4 种模型都呈现出随着边抽取比例增大,效果先增后减的趋势.这是因为抽取边比例增大,意味着测试边集合增大,所以边命中的概率也随之增大;另一方面,随着抽取边比例增大,提供给模型训练的信息也就越少,当提升测试边集合的收益已经不能抵消模型训练信息减少而带来的损失时,效果就开始下降了.



**Fig.3** Influence of ratio of removed links

**图 3** 边移除比例的影响

## 2.5 参数敏感性分析

本节测试 NEES 模型对参数的敏感性.对于第 1 阶段的边向量训练模型,我们度量了边向量维度  $d_1$  对模型的影响;对于游走阶段,则测试了节点游走次数  $r$  和游走长度  $l$  的影响;对于 Skip-Gram 模型,我们测试了节点向

量维度  $d_2$  的影响以及上下文窗口大小  $w$  的影响.除了当前被测试的参数,其他参数均保持默认值.测试任务使用 BLOGCATALOG 数据集进行多标签分类任务来展示效果.

我们首先评估维度变化对 NEES 模型的影响.结果如图 4 所示:随着维度的增加,模型的效果略微提升,这是因为更大的维度能够存储更多的信息.尤其是对于第 1 阶段的边向量训练模型,因为关系类型信息比节点信息要更丰富,所以边向量维度  $d_1$  对 NEES 模型的影响要比节点向量维度  $d_2$  的影响稍明显一些.

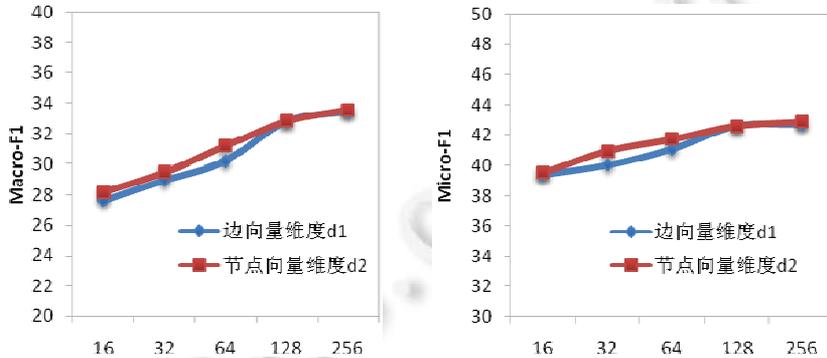


Fig.4 Influence of link vector dimension  $d_1$  and node vector dimension  $d_2$

图 4 边向量维度  $d_1$  和节点向量维度  $d_2$  的影响

另外,我们测试了游走参数(游走次数  $r$  和游走长度  $l$ )对模型的影响,结果如图 5 所示.可以看出:随着参数  $r$  的增大,模型效果首先快速提升,然后进入一个震荡区间.至于参数  $l$ ,模型的效果持续提升,但效果趋于平缓.两个参数能使 NEES 模型效果提升,是因为他们让游走阶段能够遍历图中更多的可能路径,来给模型提供更多的有用信息,但持续增加,提供信息开始变得冗余.参数  $r$  的默认值为 10,而  $l$  的默认值为 80,所以参数  $r$  能够更早地达到一个合适的值.

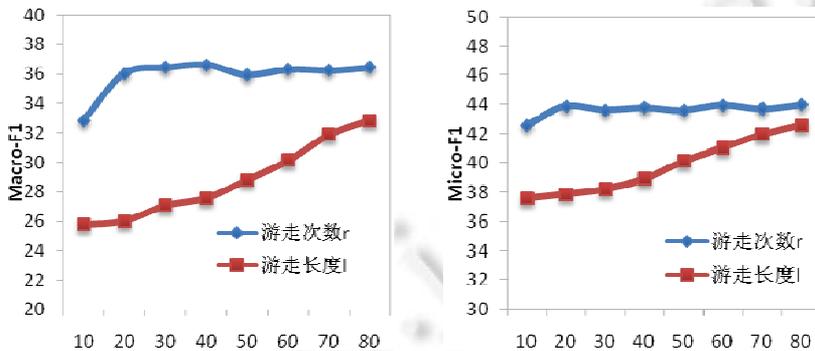


Fig.5 Influence of random walk times  $r$  and length  $l$

图 5 游走参数  $r$  和  $l$  的影响

至于窗口上下文参数  $w$  对模型的影响,我们也进行了探究.我们将参数  $w$  的值由 6 增加到 16,模型的效果只是出现了幅度不超过 1%的波动.因为随着窗口的增加,能够提供有用信息的节点和噪声数据都被囊括进来,而 Skip-Gram 模型并不关系上下文节点与中心节点的距离,导致了这种波动的产生.而在模型效率方面,模型的运行时间也会随窗口的增大而呈现线性增长的趋势.因此,结合实验效果和实验效率来看,将窗口大小设为一个较小的值是更好的选择.

## 2.6 可规模化分析

在第 2.5 节的分析和讨论中,我们已知 NEES 模型的时间复杂度可以认为是与网络的节点数呈线性相关的.本节对 NEES 模型的可规模化进行了量化分析.与文献[21]类似,我们利用开源的 python 的复杂网络编程包 networkx 生成 Erdos-Renyi 随机图(ER 图)作为模型的原始数据,ER 图的节点个数从 100 增加到 1 000 000,节点度保持为 10.

图 6 展示了 NEES 模型各阶段随着节点数增加的时间消耗增长情况.可以看出,模型的时间消耗确实与网络节点数呈线性相关.另外可以看出:其中耗时最长的是第 2 阶段的随机游走部分,与 Node2Vec 模型一样,NEES 模型也需要在游走前计算转移概率矩阵,它涉及到余弦相似度计算,是相对耗时的操作.得益于别名采样法,随机游走过程可以在  $O(1)$  时间复杂度内完成.而第 1 阶段的边向量模型和第 2 阶段的节点向量模型都使用了负采样技术和异步随机梯度下降法,提高了算法的效率.可见:这两部分耗时都相当少,尤其是第 1 阶段的边向量模型,训练速度非常快.

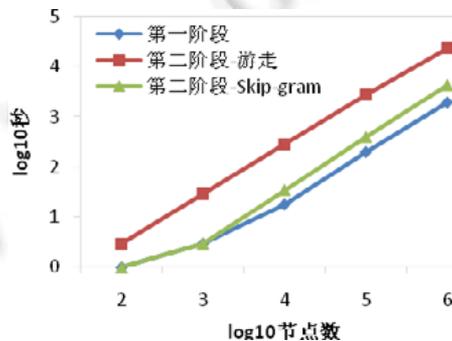


Fig.6 Training time w.r.t. number of nodes

图 6 关于不同节点数下的训练时长

## 3 总结与展望

本文提出了无监督网络表示学习模型 NEES,它不仅能够保留邻居节点信息,同时也能保存节点与邻居的关系信息.NEES 模型以 DeepWalk 模型为基础,主要对 DeepWalk 两个方面做了改进:一方面,它改进了 DeepWalk 模型的随机游走过程,提高模型对图的二阶相似性的保存的准确性;另一方面,NEES 模型对 DeepWalk 优化部分使用的 Skip-Gram 模型进行了改进,添加了它对图的一阶相似性保存的能力.通过在几个真实世界的网络上进行了多标签分类和边预测任务,证明了 NEES 模型在 3 个应用场景下都能取得优秀表现.NEES 模型的边向量模型为区分关系类型提供了一种新方式,它无需标记数据,并且是可规模化的,能够应用到真实世界的大型网络.

在下一阶段,我们将从以下几个方面进行尝试.

- (1) 结合深度学习相关理论,对网络表示训练模型进行扩展和完善.目前,大部分模型都是浅层模型,随着深信度网络、深度残差网络和 GPU 技术的推广,研究如何利用深度学习理论保存更大规模数据的信息,变得更加可行和必要;
- (2) 更好地利用和保存一阶相似性.在实验部分可以看出:一阶相似性对于稀疏网络的学习有非常重要的作用,但目前方法对此还不够重视.直接邻居对节点的指示作用非常明显,如何更好地利用该信息,是我们将来的工作方向之一;
- (3) 边向量模型在关系类型识别和知识图谱补全方面的应用研究.本文已经证明:边向量模型确实可以使得关系类型相似的边的向量相似,所以容易想到它在关系类型识别方向可能存在较大的潜力.相关方向如知识图谱补全,或许也将从中受益.但目前,本模型还没有针对如何训练合适的关系类型向量进行优化,实体之间存在多边关系也尚未涉及,这些问题都值得思考.

**References:**

- [1] Zhu P, Qian T, Zhong M, Li X. Inferring users' gender from interests: A tag embedding approach. In: Proc. of the Neural Information Processing. Springer Int'l Publishing, 2016. 86–94. [doi: 10.1007/978-3-319-46681-1\_11]
- [2] Fouss F, Pirotte A, Renders JM, Saerens M. Random-Walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. on Knowledge & Data Engineering*, 2007,19(3):355–369. [doi: 10.1109/TKDE.2007.46]
- [3] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. *ACM Computing Surveys*, 2009,41(3):1–58. [doi: 10.1145/1541880.1541882]
- [4] Chang S, Han W, Tang J, Qi GJ, Aggarwal CC, Huang TS. Heterogeneous network embedding via deep architectures. In: Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2015. 119–128. [doi: 10.1145/2783258.2783296]
- [5] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 2013,35(8):1798. [doi: 10.1109/TPAMI.2013.50]
- [6] Gallagher B, Eliassirad T. Leveraging label-independent features for classification in sparsely labeled networks: An empirical study. *Lecture Notes in Computer Science*, 2010,5498:1–19. [doi: 10.1007/978-3-642-14929-0\_1]
- [7] Henderson K, Gallagher B, Li L, Akoglu L, Eliassi-Rad T, Tong H, Faloutsos C. It's who you know: Graph mining using recursive structural features. In: Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2011. 663–671. [doi: 10.1145/2020408.2020512]
- [8] Chen WZ, Zhang Y, Li XM. Representation learning on network. *Big Data*, 2015,1(3):8–22 (in Chinese with English abstract).
- [9] Venna J, Kaski S. Local multidimensional scaling. *Neural Networks the Official Journal of the Int'l Neural Network Society*, 2006, 19(6-7):889. [doi: 10.1016/j.neunet.2006.05.014]
- [10] Tenenbaum JB, Silva VD, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000, 290(5500):2319–2323. [doi: 10.1126/science.290.5500.2319]
- [11] Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000,290(5500):2323–2326. [doi: 10.1126/science.290.5500.2323]
- [12] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Proc. of the Int'l Conf. on Neural Information Processing Systems: Natural and Synthetic. MIT Press, 2001. 585–591.
- [13] Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. *American Statistician*, 1992,46(3):175–185. [doi: 10.1080/00031305.1992.10475879]
- [14] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. *Computer Science*, 2013.
- [15] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations. In: Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2014. 701–710. [doi: 10.1145/2623330.2623732]
- [16] Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2016. 855. [doi: 10.1145/2939672.2939754]
- [17] Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. LINE: Large-Scale information network embedding. In: Proc. of the Int'l Conf. on World Wide Web. ACM Press, 2015. 1067–1077. [doi: 10.1145/2736277.2741093]
- [18] Li AQ, Ahmed A, Ravi S, Smola AJ. Reducing the sampling complexity of topic models. In: Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2014. 891–900. [doi: 10.1145/2623330.2623756]
- [19] Morin F, Bengio Y. Hierarchical probabilistic neural network language model. *Aistats*, 2005.
- [20] Bordes A, Glorot X, Weston J, Bengio Y. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 2014,94(2):233–259. [doi: 10.1007/s10994-013-5363-6]
- [21] Bordes A, Usunier N, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. In: Proc. of the Advances in Neural Information Processing Systems. 2013. 2787–2795.
- [22] Wang D, Cui P, Zhu W. Structural deep network embedding. In: Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2016. 1225–1234. [doi: 10.1145/2939672.2939753]
- [23] Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters. *ACM Trans. on Knowledge Discovery from Data*, 2006,1(1):2. [doi: 10.1145/1217299.1217301]

附中文参考文献:

- [8] 陈维政,张岩,李晓明.网络表示学习.大数据,2015,1(3):8-22.



陈丽(1992-),女,湖南衡阳人,硕士生,主要研究领域为 Web 挖掘,数据管理.



朱辉(1983-),男,工程师,主要研究领域为电力金融,互联网金融,项目管理.



朱裴松(1994-),男,博士,硕士生,主要研究领域为 Web 挖掘,数据管理.



周静(1984-),男,政工师,主要研究领域为智能电网,信息通信,大数据,项目管理.



钱铁云(1970-),女,博士,教授,博士生导师,CCF 专业会员,主要研究领域为 Web 挖掘,数据管理.

www.jos.org.cn