

DOI:10.13232/j.cnki.jnju.2019.04.007

## 基于路径相互关注的网络嵌入算法

钱付兰\*, 黄鑫, 赵姝, 张燕平

(安徽大学计算机科学与技术学院, 合肥, 230601)

**摘要:**网络嵌入, 或者称为网络表示学习, 旨在将网络中的节点映射到表示空间中, 生成低维稠密的向量, 从而在保留网络结构信息的前提下对网络中的节点进行表示, 而后通过已有的机器学习方法解决诸如链接预测、节点分类、社团发现和网络可视化等下游任务。随机游走算法可以很好地探索网络中节点的局部结构, 然而之前的基于随机游走的表示学习算法只能为节点产生一种角色嵌入, 没有考虑到和不同邻居进行交互时节点扮演的不同角色嵌入。因此, 提出一种基于路径相互关注的网络嵌入算法, 使用节点随机游走产生的上下文信息, 通过注意力机制为每个节点生成上下文相互关注的节点嵌入。在真实数据集上的实验结果表明, 与三个经典的网络嵌入算法相比, 该算法具有更好的表现。

**关键词:**网络表示学习, 随机游走, 相互关注, 注意力机制

**中图分类号:**TP391

**文献标识码:**A

## Path-based mutual attention algorithm for network embedding

Qian Fulan\*, Huang Xin, Zhao Shu, Zhang Yanping

(School of Computer Science and Technology, Anhui University, Hefei, 230601, China)

**Abstract:** Network embedding, or network representation learning, aims to map nodes in the network into the representation space and generate low-dimensional dense vectors to represent the nodes in the network while preserving the network structure information, then solve downstream tasks such as link prediction, node classification, community discovery and network visualization through existing machine learning methods. The random walk algorithm can well explore the structure of nodes in the network. However, the previous representation learning algorithm based on random walk can only generate one kind of embedding for one node, without considering that the nodes play different roles when interacting with different neighbors. Therefore, this paper proposes a network embedding algorithm based on mutual attention of paths. Through the context information generated by random walks of nodes, each node generates a node embedding in which contexts are of mutual attention. And our algorithm has better performance than the three classic network embedding algorithms.

**Key words:** network representation learning, random walk, mutual attention, attention mechanism

在信息时代, 网络是一种普遍存在的数据结构, 广泛应用于计算机科学及相关领域<sup>[1]</sup>。在

现实世界中, 网络可以表示复杂的关系系统, 比如社交网络信息和论文引用网络信息。以社交

基金项目: 国家重点研究与发展项目(2017YFB1401903), 国家自然科学基金(61673020, 61702003, 61876001), 安徽省自然科学基金(1808085MF175)

收稿日期: 2019-05-28

\* 通讯联系人, E-mail: qianfulan@hotmail.com

系统为例,个体被映射成网络中的节点,个体之间的社交关系被映射成网络中的边,从而将社交系统抽象为网络,通过数据挖掘等技术进一步分析系统的信息.邻接矩阵是网络数据的简单直接表示,其每一行表示一个节点和所有其他节点之间的连边,可以视为对应节点的表示<sup>[2]</sup>,但由于信息网络可以包含数十亿个节点和边,因此在整个网络上执行复杂的推理过程是十分困难的<sup>[3]</sup>.

网络嵌入中的网络表示学习,旨在学习潜在的、低维的节点表示,同时保持网络拓扑结构<sup>[4]</sup>.网络嵌入提供了一种高效的方法,将网络中的信息保留,从而减轻了传统方法由于稀疏性所带来的问题求解困难.因此,近年来,网络嵌入吸引了越来越多的关注,在链路预测、节点分类和聚类网络相关问题分析任务上取得了良好的性能<sup>[5]</sup>.作为热门的研究课题,已经有很多网络表示学习模型被提出.大多数先前的嵌入算法都采用对每个节点生成一种角色嵌入,没有考虑到和不同邻居进行交互时节点扮演的不同角色嵌入<sup>[6]</sup>.举例来说,当一篇论文A被两篇论文B和C引用时,在引文网络中节点A与节点B,C分别有边相连,一般的处理方式往往认为B,C较为相似,而实际情况中B可能是自然语言处理方向的论文,C可能是网络嵌入方向的论文,B和C未必相似.因此对节点A进行节点嵌入时,需要考虑到与不同邻居交互时扮演的这种角色差别.最近,随着深度学习和自然语言处理技术的发展,又出现了许多通过节点的文本信息等其他信息的网络表示学习方法.在矩阵分解的框架下,TADW(Text-Associated DeepWalk)<sup>[7]</sup>将节点的文本特征引入网络表示学习中.CENE(Content-Enhanced Network Embedding)<sup>[8]</sup>通过将内容信息视为一种特殊的节点,将文本内容合并到网络表示学习的过程中.这些算法可以显著地提升网络表示学习的效果,但是网络本身的信息可能未被充分地挖掘.另外,并不是所有的现实世界网络都具有文本信息,而某些文本信息的稀疏性或

者噪声信息不足以使带有文本信息的网络具有更好的表示学习效果,所以基于网络结构的网络表示学习仍然具有重要的意义.为了解决上面的问题,本文提出了一种基于路径注意力机制的网络表示学习算法PMANE(Path-based Mutual Attention for Network Embedding).本文的主要贡献可归纳为以下两个方面:

(1)算法对节点在网络中进行随机游走获取结构上下文信息,再通过注意力机制获取不同邻居节点的相互关注信息,有效区分了节点在和不同邻居节点交互时的角色,从而获取了较为准确的网络嵌入表示.

(2)算法针对网络结构本身进行数据挖掘,通过改进的随机游走策略和注意力机制有效解决了文本获取困难或不足时,如何挖掘网络结构更深层次的信息的问题.

## 1 相关工作

早期的网络表示学习大部分是基于矩阵分解思想,矩阵分解算法主要通过表示网络中节点的连接关系,然后对矩阵进行分解从而获得节点的向量表示.最早基于矩阵分解的网络表示学习算法LLE(Locally Linear Embedding)由Roweis and Saul<sup>[9]</sup>提出,认为在向量空间中的节点表示应该是该节点所有邻居节点的线性组合,并且权重由节点的邻接矩阵所确定.Laplace特征表示<sup>[10]</sup>则认为如果两个节点相连,那么它们的表示应该接近,并且这种接近程度是用欧式距离来衡量的.SDNE(Structural Deep Network Embedding)<sup>[11]</sup>通过深层神经网络对节点进行表示学习,首先使用Laplace矩阵进行第一级相似度的建模,而后使用深层自编码器建模二级相似度.随着自然语言处理技术的发展,Mikolov et al<sup>[12]</sup>于2013年提出了word2vec模型,这是一种用于学习词表示的神经网络模型,并取得了良好的效果.Perozzi et al<sup>[13]</sup>观测到节点在短随机游走中的分布和词语在自然语言中的分布都满足幂律分布,从而将word2vec模型引入网络表示学习,提出Deep-

walk算法,使得效果有了较大的提升. Tang et al<sup>[14]</sup>随后又提出LINE(Large-scale Information Network Embedding)算法,使用简单的神经网络学习网络表示,通过一阶邻近和二阶邻近来刻画节点之间的关系. 其中直接相连的一对节点表示这对节点有着一阶邻近关系,不相邻的一对节点的共同邻居刻画了这对节点的二阶邻近关系. Grover and Leskovec<sup>[15]</sup>于2016年提出了改进DeepWalk的算法node2vec,通过引入参数 $p$ 和 $q$ 调节随机游走的深度和广度的偏好关系提升了算法的性能.

随着卷积神经网络(Convolutional Neural Networks, CNN)的提出以及数值计算设备的不断改进,越来越多的算法效果通过运用CNN得到了提升,特别是在图像领域和自然语言处理领域. Kalchbrenner et al<sup>[16]</sup>提出采用卷积神经网络用于句子建模. 同年, Hu et al<sup>[17]</sup>提出了用于匹配自然语言句子的卷积神经网络体系结构. 2016年 Attentive Pooling Networks<sup>[18]</sup>被提出用于问答系统,取得了很好的效果. Tu et al<sup>[6]</sup>于2017年提出的CANE(Context-Aware Network Embedding)认为在真实世界中一个节点可能在和不同的节点交互时表现出不同的特性,从而利用节点的文本信息进行相互关注的网络表示学习,并采用Attentive Pooling模型,取得了较好的效果. 但是CANE只能应用于具有有效文本信息的网络.

综上所述,一般的随机游走算法如DeepWalk只能对一个节点产生一种嵌入,无法捕获不同邻居进行交互节点间相互关注信息,常见的使用文本信息提升网络表示算法性能,文本的获取存在一定的困难. 受CANE算法启发,本文提出一种适用于同质网络中的路径相互关注的网络嵌入算法PMANE. 该算法关注的对象并不是节点的属性(比如文本信息),而是网络本身的结构信息. 通过对结构信息进行深入的挖掘,实现在不带有属性信息的网络中进行表示学习也能取得较好的实验效果. 由于短游走序列中的词序列和自然语言中的词都服从

律分布,因而可以用CNN来对短游走所得的节点序列进行卷积. 相较于某些传统的随机游走算法,这里获取的是一种更加高阶的信息,更加类似于DeepWalk算法. 同时,PMANE加入注意力机制,使相同节点对于不同的邻居节点有着不同的嵌入. PMANE针对链路预测问题在三个公共数据集上进行了实验,实验结果相较于传统方法获得了较大的提升,意味着PMANE获得了更精确的表示.

## 2 问题定义和算法描述

本节给出网络表示学习问题形式化定义,并且介绍相关公式.

**2.1 基本定义** 首先给出基本的符号和定义. 假定有一个网络 $G=(V, E)$ ,其中 $V$ 是节点集, $E \subseteq V \times V$ 是节点之间的边集, $e_{u,v} \in E$ 表示节点 $u$ 和节点 $v$ 之间的关系. 网络表示学习旨在通过网络的结构信息,对于每一个节点 $v \in V$ 学习到一个低维嵌入向量 $v \in R^d$ ,并且, $d \ll |V|$ 是表示空间的维度.

**2.2 随机游走策略** 本文针对有向图和无向图使用不同的随机游走策略. 以图1中的无向图为例,节点A同时和节点B以及节点C有连边. 在无向图中,PMANE采用随机游走的方法产生一系列的长度为 $L$ 的短游走序列并且这些序列的中心节点为主节点. 如图1所示,以A节点为中心节点的游走序列和以B节点及C节点为中心节点的游走序列共享不同的邻居节点,PMANE加入的注意力机制有机会捕获到这种游走产生的节点上下文信息,从而加强对节点的表示. 在有向图中将初始节点作为主节点,依然采用随机游走的方法,但有向图中的随机游走容易出现断链的情况,即:未能游走至足够长度以产生有效的上下文信息. 本文采用另一种随机游走的策略,若有向图中若游走至无出度节点,将返回至初始节点继续随机游走,这种策略不仅可以克服有向图中游走长度可能过短的问题,而且有机会保留初始节点的广度优先搜索信息,更好地进行网络的表示学习.

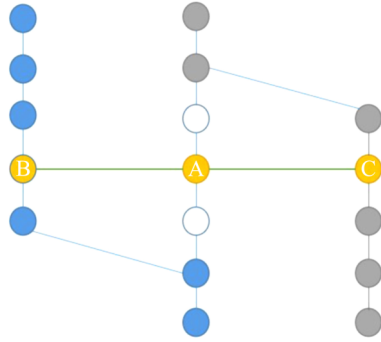


图1 基于路径的相互关注机制

Fig. 1 Path-based mutual attention mechanism

**2.3 注意力机制** 总体来说,注意力机制使池化层能知道当前的输入节点对,来自两个输入节点的上下文节点信息可以直接影响对方表示的计算,随后可以为每个节点产生相应的相互关注向量.具体流程将在第三节中详细介绍.

### 3 基于路径相互关注算法

本节详细介绍本文提出的基于路径注意力机制的网络表示学习算法,算法主要利用注意力机制对网络中一条边上的两个节点及以这两个节点产生的节点序列进行路径相互关注的节点嵌入.

为了更好地保留网络中的信息,使用了两种类型的嵌入,一种是保留一阶邻近信息的嵌入向量  $v^l$ ,第二种是保留高阶相互关注信息的  $v^h$ ,可以把它们进行连接,得到节点的嵌入  $v = v^l \oplus v^h$ ,这里的  $\oplus$  表示向量的级联操作.

**3.1 全局目标函数** PMANE 旨在最大化总体目标函数,定义如式(1)所示:

$$O = \sum_{e \in E} L(e) \quad (1)$$

其中,  $L(e)$  是由两部分组成的,如式(2):

$$L(e) = L_l(e) + L_h(e) \quad (2)$$

$L_l(e)$  表示基于低阶邻近的目标函数,  $L_h(e)$  表示基于高阶相互关注的目标函数.接下来将详细阐述这两种目标函数.

**3.2 基于低阶邻近的目标函数** 不失一般性的,假设网络是有向的,因为无向边可以看作是方向相反、权值相等的有向边.因此,基于

低阶邻近的目标旨在使用基于低阶邻近的嵌入来测量定向边缘的对数似然,定义如式(3):

$$L_l(e) = w_{u,v} \lg p(v|u^l) \quad (3)$$

并且,根据 LINE 提到的方法,定义了由  $u$  产生  $v$  的条件概率,如式(4)所示:

$$p(v|u^l) = \frac{\exp(u^l \cdot v^l)}{\sum_{z \in V} \exp(u^l \cdot z^l)} \quad (4)$$

**3.3 基于高阶相互关注的目标函数** 通过随机游走生成节点的路径序列信息产生的序列保存了网络结构的高阶信息.不同于 DeepWalk 所使用的 Skip-gram 模型,PMANE 用 CNN 去获取节点序列的信息.如图 2 所示,在 PMANE 中,将节点序列作为输入,采用 looking-up 层、卷积层以及池化层得到节点相应的嵌入.

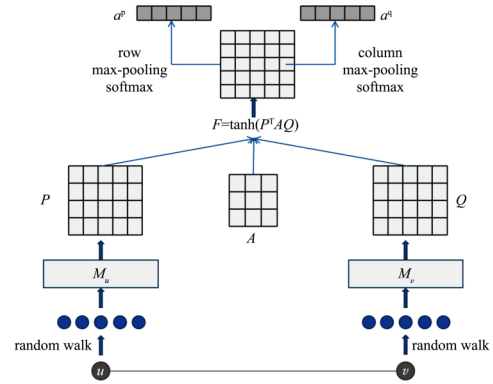


图2 PMANE 算法框架

Fig. 2 The framework of PMANE algorithm

具体的算法流程如算法 1.

算法 1 PMANE

输入: 网络  $G=(V, E)$ , 相关的步长  $t$  等参数

输出: 节点上下文感知嵌入

1. for  $v \in V$  do:
2. 生成随机游走序列  $S_v = (N_1, N_2, \dots, N_t)$
3. 初始化一阶嵌入向量  $v^l$
4. 初始化作为上下文的节点的嵌入向量  $v^h$
5. end for
6. 初始化相互关注矩阵  $A$
7. for  $e_{u,v} \in E$  do:
8. for  $(u, v) \in e$  do:
9. 通过  $S_v$  索引  $v^h$  生成矩阵  $M_u, M_v$
10. 对于  $M_u, M_v$  卷积生成矩阵  $P, Q$
11. 由式(6)生成矩阵  $F$



12. 由式(7)和式(8)生成向量  $\mathbf{a}^p$  和  $\mathbf{a}^q$
13. 由式(9)生成高阶感知向量  $\mathbf{u}_v^h, \mathbf{v}_u^h$
14. 由式(1)进行训练
15. end for
16. 由式(12)结合低阶向量和高阶感知向量得到每条边的上下文感知向量  $\mathbf{u}_v, \mathbf{v}_u$
17. end for
18. return 上下文感知向量

looking-up 层:通过随机游走的方式,生成一系列的节点序列,在无向图中,这些序列的中心节点为对应的主节点,在有向图中,以初始节点为主节点. 故而可以得到  $S = (N_1, N_2, \dots, N_l)$ , 在 looking-up 层中,将节点序列  $S$  中的每个节点  $N_i$  转化为对应的向量表示  $N_i \in R^d$ , 这里的  $d$  表示嵌入向量的维度.

卷积层:经过 looking-up 层,节点序列  $S$  中的节点通过索引对应的向量生成相应的矩阵,通过采用适当的补零操作,PMANE 使用 CNN 对嵌入序列  $S$  进行特征提取,使用卷积矩阵  $\mathbf{C}$  在长度  $l$  的滑动窗口上执行卷积运算,如式(5)所示:

$$x_i = \mathbf{C} \cdot S_{i:i+l-1} + b \quad (5)$$

如上文所说,节点在和不同的邻居交互时往往扮演不同的角色,所以 PMANE 引入注意力机制去获得这种节点相互关注的信息. 与 CANE 算法类似,attentive pooling 算法中注意力机制被用于问答系统. 由于随机游走生成的节点和信息处理中的单词同样遵循幂律分布,故而产生的节点序列也可以运用注意力机制.

在图 2 中,给出了路径感知嵌入的生成过程. 给定一条边  $e_{u,v}$ ,其中以  $u$  为主节点有一条路径,以  $v$  为主节点也有一条路径,可以通过卷积操作获得对应的矩阵  $\mathbf{P}$  和矩阵  $\mathbf{Q}$ ,接下来引入一个随机初始化的相互关注矩阵  $\mathbf{A}$ ,计算相关矩阵  $\mathbf{F}$  如式(6)所示:

$$\mathbf{F} = \tanh(\mathbf{P}^T \mathbf{A} \mathbf{Q}) \quad (6)$$

而后,对矩阵  $\mathbf{F}$  的行和列分别进行 max-pooling 操作. 与 mean-pooling 相比,在我们的实验中, max-pooling 有更好的实验结果. pooling 操作如式(7)所示:

$$\begin{aligned} g_i^p &= \max(F_{i,1}, \dots, F_{i,n}) \\ g_i^q &= \max(F_{1,i}, \dots, F_{n,i}) \end{aligned} \quad (7)$$

通过上述操作,便可获得  $\mathbf{P}$  和  $\mathbf{Q}$  的重要性向量.

PMANE 使用 softmax 函数将重要性向量转化为注意力向量  $\mathbf{a}^p$  和  $\mathbf{a}^q$ ,举例说明,如式(8):

$$\mathbf{a}_i^p = \frac{\exp(g_i^p)}{\sum_{j \in [1,n]} \exp(g_j^p)} \quad (8)$$

从而,如式(9)所示,PMANE 的  $u$  和  $v$  的节点感知嵌入向量为:

$$\begin{aligned} \mathbf{u}_{(v)}^h &= \mathbf{P} \mathbf{a}^p \\ \mathbf{v}_{(u)}^h &= \mathbf{Q} \mathbf{a}^q \end{aligned} \quad (9)$$

从而,  $L_h(e)$  如式(10)所示:

$$L_h(e) = \alpha \cdot L_{hh}(e) + \beta \cdot L_{hl}(e) + \gamma \cdot L_{lh}(e) \quad (10)$$

其中,  $\alpha, \beta$  和  $\gamma$  是超参,控制着条件概率,如式(11)所示:

$$\begin{aligned} L_{hh}(e) &= w_{u,v} \lg p(\mathbf{v}_{(u)}^h | \mathbf{u}_{(v)}^h) \\ L_{hl}(e) &= w_{u,v} \lg p(\mathbf{v}_{(u)}^h | \mathbf{u}_{(v)}^l) \\ L_{lh}(e) &= w_{u,v} \lg p(\mathbf{v}_{(u)}^l | \mathbf{u}_{(v)}^h) \end{aligned} \quad (11)$$

**3.4 级联嵌入向量** 在这一步中,一个节点的嵌入向量便可以表示为低阶邻近嵌入向量和高阶相互关注的嵌入向量的级联,如式(12)所示:

$$\begin{aligned} \mathbf{u}_{(v)} &= \mathbf{u}^l \oplus \mathbf{u}_{(v)}^h \\ \mathbf{v}_{(u)} &= \mathbf{v}^l \oplus \mathbf{v}_{(u)}^h \end{aligned} \quad (12)$$

**3.5 算法优化** 由于 softmax 的计算成本过高,所以 PMANE 需要采用负采样,从而使得算法的效率更高. 给出负采样算法,如式(13):

$$\lg \sigma(\mathbf{u}^T \cdot \mathbf{v}) + \sum_{i=1}^k E_{z \sim P(v)} [\lg \sigma(-\mathbf{u}^T \cdot \mathbf{z})] \quad (13)$$

其中,  $k$  是负样本数,  $\sigma$  表示 sigmoid 函数,  $p(v)$  表示节点的分布,其中  $d_v$  是节点  $v$  的出度. 并使用 Adam<sup>[19]</sup> 进行优化.

## 4 实验与结果

为验证提出的 PMANE 算法性能,选取三个真实的数据集进行实验结果对比和分析. 为了方便与基准算法进行比较,将所有的下游任务都设置为链路预测任务. 在所有的方中,将嵌入向量的维度都设置为 200.

**4.1 数据集** Cora 是一个典型的引文网络数

据集,由 McCallum et al<sup>[20]</sup>构建,其中含有 2277 篇机器学习论文,分为七类,为有向图.

HepTh(高能物理理论)是由 Leskovec et al<sup>[21]</sup>构建的引文网络数据集,保留 1038 篇论文,为有向图.

NetScience<sup>[22]</sup>是一个科学合作网络数据集,包含 1461 个节点以及 2742 条边,为无向图.三个数据集的信息如表 1 所示.

表 1 三个真实网络

Table 1 Three real-world networks

数据集	Cora	HepTh	NetScience
Vertices	2277	1038	1461
Edges	5214	1990	2742
Type	directed	directed	undirected
Labels	7	—	—

**4.2 对比算法设置** DeepWalk 是一种基于结构的网络嵌入模型,它使用随机游走策略,并且运用了自然语言处理中的 Skip-gram 模型,从而生成节点的嵌入,设置的游走步长为 80.

LINE 通过利用一阶和二阶信息,学习大规模网络中的顶点嵌入.实验中将 LINE 的负样本数设置为 5.

node2vec 采用带有偏置的随机游走,相对于 DeepWalk 的完全随机游走,node2vec 可以有偏置地进行游走,其中  $p$  和  $q$  两个超参是需要进行额外训练的.

**4.3 实验环境及参数设置** PMANE 在数据集上设置游走步长为 6,训练迭代次数为 300 次,相互关注矩阵  $A$  的维度为  $100 \times 100$ ,学习率初始为 0.1,每迭代一次变为原来的 0.95 倍.

实验硬件环境为:NVIDIA GeForce GTX 860 M GPU, Intel Core i5-4200H (2.8 GHz  $\times$  4) CPU, 8 G 内存.

软件环境为:Windows10 系统,使用 python 3.6 版本,主要 python 第三方包及版本有:numpy 1.14.5 版, networkx 2.2 版, tensorflow 1.12.0 版.

**4.4 实验结果与分析** 链路预测的目标是预

测网络中没有出现的边或将来可能出现的边.对于链路预测任务采用标准的衡量指标  $AUC$  (Area Under a receiver operating characteristic (roc) Curve),它表示随机未被观测链路中的顶点比随机不存在的链路中的顶点对更相似的概率.在网络嵌入中,一般采用一对顶点的表示向量的余弦相似度或者向量内积来计算得分,本文中采用向量内积来计算得分,  $AUC$  的值大于 50% 的程度越高,表示算法的性能越好.将划分比例  $r$  从 15% 调整到 95% 进行计算.由于  $r=5\%$  时,训练集中大多数的点都是孤立的,所以没有进行 5% 训练集下的实验.按照不同比例分别随机去除 Cora, HepTh 和 NetScience 中的边作为测试集,保留的边作为训练集进行训练,其预测结果计算  $AUC$  值分别如表 2、表 3 和表 4 所示,表中黑体字是表现最优的结果.

对 Cora 数据集和 HepTh 数据集进行了不同比例下 PMANE 对于对比算法的提升比例,这样可以直观地看出在不同比例下 PMANE 算法性能.如图 3 所示.

可以观察到几点信息:

(1)从表 2、表 3 和表 4 中,与传统算法对比可以观察到本文提出的 PMANE 模型在不同数据集、不同比例训练集上,与所有基准算法相比,均取得了一定程度的提升.算法在链路预测上的有效性说明了 PMANE 可以准确地对节点进行表示.

(2)通过观察图 3,可以发现,PMANE 在较小的划分比例下对节点的表示效果较好.因为在数据集较为稀疏时,算法通过随机游走产生的节点序列可以很好地探索网络的结构,使得注意力机制可以更好地捕获节点的路径信息.

## 5 总 结

本文通过对网络中的节点进行随机游走,并对有向图中的随机游走进行改进,然后使用注意力机制对每个节点的嵌入进行强化,从而增强了表示学习的效果,挖掘了网络结构更深层次的信息.在三个数据集上的相关实验也验

表2 Cora数据集下AUC值

Table 2 AUC on Cora

训练比例	15%	25%	35%	45%	55%	65%	75%	85%	95%
Deepwalk	56.0	63.0	70.2	75.5	80.1	85.2	85.3	87.8	90.3
LINE	55.0	58.6	66.4	73.0	77.6	82.8	85.6	88.4	89.3
Node2vec	55.9	62.4	66.1	75.0	78.7	81.6	85.9	87.3	88.2
PMANE	<b>62.2</b>	<b>66.4</b>	<b>70.3</b>	<b>78.2</b>	<b>81.9</b>	<b>86.7</b>	<b>88.0</b>	<b>90.0</b>	<b>90.5</b>

表3 HepTh数据集下AUC值

Table 3 AUC on HepTh

训练比例	15%	25%	35%	45%	55%	65%	75%	85%	95%
Deepwalk	55.2	66.0	70.0	75.7	81.3	83.3	87.6	88.9	88.0
LINE	53.7	60.4	66.5	73.9	78.5	83.8	87.5	87.7	87.6
Node2vec	57.1	63.6	69.9	76.2	84.3	87.3	88.4	89.2	89.2
PMANE	<b>59.6</b>	<b>67.0</b>	<b>74.3</b>	<b>81.5</b>	<b>85.2</b>	<b>89.4</b>	<b>89.5</b>	<b>91.1</b>	<b>92.9</b>

表4 NetScience数据集下AUC值

Table 4 AUC on NetScience

训练比例	15%	25%	35%	45%	55%	65%	75%	85%	95%
Deepwalk	80.6	83.5	90.1	94.5	95.3	97.4	98.2	98.4	97.6
LINE	77.8	81.5	88.4	92.0	94.2	96.8	97.8	98.4	98.7
Node2vec	80.5	85.5	89.8	93.8	95.0	97.3	97.4	98.0	97.2
PMANE	<b>81.8</b>	<b>85.6</b>	<b>91.8</b>	<b>94.6</b>	<b>96.2</b>	<b>97.5</b>	<b>98.3</b>	<b>98.6</b>	<b>98.8</b>

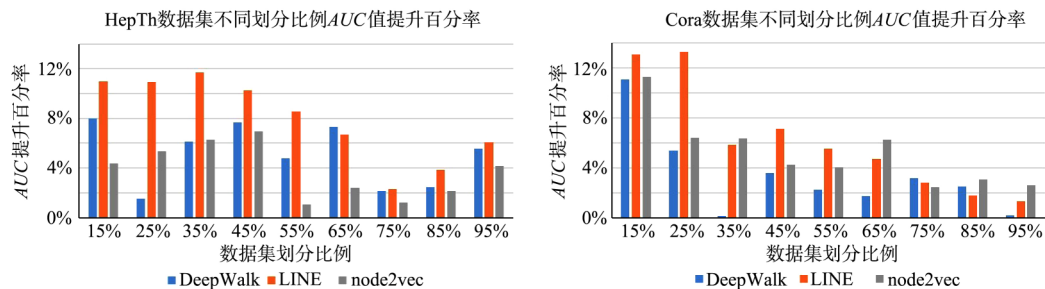


图3 不同比例下PMANE对基线算法的提升百分率

Fig. 3 Percentage increase of PMANE to baseline algorithm at different ratios

证了本文算法的有效性. 未来将使用这个方法  
和文本信息、图像信息等外部信息融合, 以增强  
嵌入的效果.

#### 参考文献

- [1] Hamilton W L, Ying R, Leskovec J. Representation learning on graphs: methods and applications. 2017, arXiv:1709.05584.
- [2] 涂存超, 杨成, 刘知远等. 网络表示学习综述. 中国科学: 信息科学, 2017, 47(8): 980—996. (Tu C C, Yang C, Liu Z Y, et al. Network representation learning: an overview. Scientia Sinica (Informationis), 2017, 47(8): 980—996.)
- [3] Chen H C, Perozzi B, Al-Rfou R, et al. A tutorial on network embeddings. 2018, arXiv:1808.02590.

- [4] Zhang D K, Yin J, Zhu X Q, et al. Network representation learning: a survey. *IEEE Transactions on Big Data*, 2018, doi: 10.1109/TBDATA.2018.2850013.
- [5] Cai H Y, Zheng V W, Chang K. A comprehensive survey of graph embedding: problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2018, 30(9): 1616—1637.
- [6] Tu C C, Liu H, Liu Z Y, et al. Cane: context-aware network embedding for relation modeling// *Proceedings of the 55<sup>th</sup> Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada: Association for Computational Linguistics, 2017: 1722—1731.
- [7] Yang C, Liu Z Y, Zhao D L, et al. Network representation learning with rich text information// *Proceedings of the 24<sup>th</sup> International Conference on Artificial Intelligence*. Buenos Aires, Argentina: AAAI Press, 2015: 2111—2117.
- [8] Sun X F, Guo J, Ding X, et al. A general framework for content-enhanced network representation learning. 2016, arXiv: 1610.02906.
- [9] Roweis S T, Saul L K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000, 290(5500): 2323—2326.
- [10] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering// *Proceedings of the 14<sup>th</sup> International Conference on Neural Information Processing Systems*. Vancouver, Canada: MIT Press, 2001: 585—591.
- [11] Wang D, Cui P, Zhu W. Structural deep network embedding// *Proceedings of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, CA, USA: ACM, 2016: 1225—1234.
- [12] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality// *Proceedings of the 26<sup>th</sup> International Conference on Neural Information Processing Systems*. Lake Tahoe, NV, USA: MIT Press, 2013: 3111—3119.
- [13] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations// *Proceedings of the 20<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2014: 701—710.
- [14] Tang J, Qu M, Wang M Z, et al. Line: large-scale information network embedding// *Proceedings of the 24<sup>th</sup> International Conference on World Wide Web*. Florence, Italy: ACM, 2015: 1067—1077.
- [15] Grover A, Leskovec J. node2vec: scalable feature learning for networks// *Proceedings of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, CA, USA: ACM, 2016: 855—864.
- [16] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. 2014, arXiv: 1404.2188.
- [17] Hu B T, Lu Z D, Li H, et al. Convolutional neural network architectures for matching natural language sentences// *Proceedings of the 27<sup>th</sup> International Conference on Neural Information Processing Systems*. Montreal, Canada: MIT Press, 2014: 2042—2050.
- [18] Dos Santos C, Tan M, Xiang B, et al. Attentive pooling networks. 2016, arXiv: 1602.03609.
- [19] Kingma D P, Ba J. Adam: a method for stochastic optimization. 2014, arXiv: 1412.6980.
- [20] McCallum A K, Nigam K, Rennie J, et al. Automating the construction of internet portals with machine learning. *Information Retrieval*, 2000, 3(2): 127—163.
- [21] Leskovec J, Kleinberg J, Faloutsos C. Graphs over time: densification laws, shrinking diameters and possible explanations// *Proceedings of the 11<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. Chicago, IL, USA: ACM, 2005: 177—187.
- [22] Newman M E J. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 2006, 74(3): 036104, doi: 10.1103/PhysRevE.74.036104.

(责任编辑 杨可盛)