计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(9):247−255 [doi: 10.15888/j.cnki.csa.008172]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于双视角的耦合网络表示学习算法①

倪琦瑄[1], 张　霞[2], 卜　湛[3]

[1](南京财经大学 信息工程学院, 南京 210003)
[2](南京中医药大学 人工智能与信息技术学院, 南京 210023)
[3](南京财经大学 江苏省电子商务重点实验室, 南京 210003)
通讯作者: 卜　湛, E-mail: zhanbu@nufe.edu.cn

摘　要: 传统网络表示学习算法大多依赖于节点视角下的随机游走获取网络局部采样序列, 再通过最大化相邻节点的共现概率将网络中的节点表示成低维向量. 本文在真实网络上的经验分析表明, 对节点和边两种视角分别进行随机游走会产生具有不同节点分布的采样序列, 进而得到不同的社区划分. 为此, 本文提出了一种基于双视角的耦合表示学习算法 DPBCNE. 该方法基于边视角进行随机游走以获得不同于节点视角的采样结果, 再融合基于节点视角下的节点采样序列进行耦合训练, 以学习节点和边的表示. 实验结果表明, 相较于现有的网络表示学习算法, DPBCNE 能更好地保留网络拓扑结构信息, 并在下游分类和预测任务中获得更好的效果.

关键词: 双视角; 网络表示学习; 随机游走; 边图

引用格式: 倪琦瑄,张霞,卜湛.基于双视角的耦合网络表示学习算法.计算机系统应用,2021,30(9):247–255. http://www.c-s-a.org.cn/1003-3254/8172.html

Coupled Network Embedding Method Based on Dual Perspectives

NI Qi-Xuan[1], ZHANG Xia[2], BU Zhan[3]

[1](College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210003, China)
[2](College of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine, Nanjing 210023, China)
[3](Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing 210003, China)

Abstract: Traditional network embedding approaches rely heavily on random walk in a node perspective to get the local sampling sequence of networks and then maximize the co-occurrence probability between adjacent nodes to represent nodes as low-dimensional vectors. The empirical analysis of this study on a real-world network shows that random walk in node and link perspectives can respectively produce network sampling results with different node frequency distributions, resulting in various partitions of the network. To this end, this study proposes an approach to Dual Perspective Based Coupled Network Embedding (DPBCNE). DBPCNE gets the network sampling sequences by random walk in a link perspective and then combines node sequences sampled in a node perspective for coupled training. Experiments show that compared with other network embedding approaches, this approach can well preserve network structures and improve the effectiveness of network embedding for the downstream classification and prediction tasks.

Key words: dual perspective; network embedding; random walk; line graph

1　引言

现实世界中许多的社会, 物理和信息系统都是以复杂网络的形式存在的[1]. 在这些网络中, 人们通过挖掘网络的潜在信息可以解决许多实际的问题, 例如链

① 基金项目: 国家重点研发计划 (2019YFB1405000); 国家自然科学基金 (71871109)
Foundation item: National Key Research and Development Program of China (2019YFB1405000); National Natural Science Foundation of China (71871109)
收稿时间: 2021-02-01; 修改时间: 2021-02-24; 采用时间: 2021-03-11; csa 在线出版时间: 2021-09-02

路预测[2], 链路重构[3], 节点分类[4], 边分类等. 网络表示学习将节点映射到潜在空间[5], 并保留了丰富的结构信息, 为这些下游数据挖掘任务提供了一种新的解决方法, 已在许多论文中被证明能够非常有效地解决上述任务[6–13].

近 10 年里, 有很多能够保持网络的结构特性的方法被提出来. 然而, 以往的研究往往只注重节点视角下的网络拓扑信息, 而没有充分考虑边视角. 现有的网络表示学习算法可以分为两大类: 基于随机游走的网络表示学习算法和基于深度学习框架的网络表示学习算法. 基于随机游走的网络表示学习算法, 例如 DeepWalk[8] 和 Node2Vec[9], 主要分为随机游走和 Skip-gram[10] 算法两部分, 他们通过随机游走以获取节点视角下的网络拓扑结构, 再利用 Skipgram 模型对每个节点进行更新学习, Node2Vec 在 DeepWalk 的基础上, 加入了带有偏置的随机游走, 使得其能以不同偏好的游走方式获取节点的同质性和结构等价性. Line[11] 也是一种基于邻域相似假设的网络表示学习算法, 通过设置两种邻近性构造目标函数来获取节点的局部相似性和邻居相似性. 然而, 这些算法往往只关注了节点视角下网络拓扑信息, 而没有充分考虑边视角. 基于深度学习的网络表示算法, 如 SDNE[12], 利用半监督的自动编码器模型来学习网络中每个节点的表示向量, DNGR[13] 则通过 Random Surfing 模拟随机游走过程获取网络拓扑信息, 再利用堆叠降噪自编码器对节点表示向量进行训练. GAE[14] 通过图卷积作为自编码器的编码方式, 为每个节点聚合其邻居的特征信息, 再利用解码器重构网络的邻接信息. 这些基于深度学习架的网络表示学习算法往往效果很好, 但其涉及到许多复杂计算, 其空间和时间复杂度随着网络规模的扩大面临着巨大的挑战, 同时, 这些算法也只关注了节点视角下的节点之间的连接信息, 而没有考虑过边视角. 而在已有的针对边的一些研究中, 大多考虑的是边上的属性或标签信息[15,16]. 如 NEES[15] 通过边计算节点之间关系的相似性, 从而设置带有偏好的随机游走, 使得节点的随机游走的每一步都与上一步有尽可能相似的关系. ELAINE[16] 则通过重构每种边的标签向量, 使得最终学习得到的节点向量能够同时包含边的标签信息和网络结构信息. 事实上, 不同的视角会带来不同的有效信息, 比如不同的社区划分. 因此, 本文将研究重点从节点转移到边. 利用边图 (line graph)[17] 从边的角度观察网络. 在边图中, 原始网络的边表现为边图中的节点, 边图中的边表示原始网络中边之间的连接关系. Ahn 等人[18] 在对重叠社区发现问题的研究中就提出了边社区的概念, 他们通过边而不是节点来划分社区, 并指出边的层次结构与原始网络中的节点层次结构有所不同, 从边的角度观察目标网络可以发现更多的特征.

一个好的网络表示学习算法应该能够有效地保存网络中的社区结构[9], 为了达到这一效果, 现有的基于随机游走的网络表示学习方法如 DeepWalk 主要采用随机游走获取网络中每个节点的上下文信息, 继而采用 Skip-gram 模型学习其低维表示, 从而使得具有相似拓扑属性的节点具有相似的低维表示. 过往的研究表明[19], 一个社区的内部连接应该多于其外部连接, 由于社区内的连接密度应该比较高, 所以随机游走时停留在自己社区内节点上的概率要高于到外部的概率. 为了进一步了解边图, 本文选择 DBLP 数据集[20], 分别在原始网络和对应的边图上进行随机游走. 在设置相同的总采样次数的情况下, 在图 1 中画出了其不同视角下的节点采样频率分布, 横坐标表示原始网络中的节点在不同视角下的采样过程中的出现次数, 纵坐标表示对应出现次数的节点数量, 可以看到在两种视角下随机游走得到的节点频率分布是不同的, 这是因为边图可以为我们提供更多的连接信息, 进而可以为我们更好地揭示原始网络中的层次结构.

综上所述, 本文提出了一种融合节点视角和边视角的耦合网络表示学习算法 DPBCNE (Dual Perspectives Based Coupled Network Embedding). 分别从两个视角下学习网络拓扑信息, 在这个耦合模型中, 可以同时学习节点和边的低维表示. 通过使用节点-边的耦合训练机制, 可以将节点和边映射到相同的低维空间中. 与大多数现有的网络表示学习方法间接获取边的表示方法不同, DPBCNE 可以直接学习得到节点和边的表示, 更具有解释性. 本文在 4 个真实复杂网络中验证了 DPBCNE 的性能, 并通过节点分类, 边分类, 链路预测和链路重构 4 个任务来比较 DPBCNE 与当下最先进的网络表示学习方法的效果. 在链路预测任务中, 与基准方法相比, DPBCNE 取得了较好的结果, 在合作者网络中仅次于 CN 算法. 而在节点分类, 边分类和链路重构任务中, DPBCNE 均优于其他所有网络表示学习基准方法.

2 基于双视角的耦合网络表示学习算法

本文提出了一种新的基于双视角的耦合网络表示学习算法. 其主要模型框架如图 2 所示, 模型可以分为两个部分: 第 1 部分, 获取两种视角下的网络结构信息, 给定一个原始网络, 首先构建边图, 在原始网络和边图上分别进行随机游走, 得到不同视角下的节点采样序列. 第 2 部分, 耦合更新原始网络中的节点向量和边向量, 根据节点和边之间的对应关系以及不同视角下中心词节点和其上下文节点的关系, 将上述两个视角结合起来进行耦合更新, 在更新过程中共享节点和边的向量表示, 最终获得融合两种视角下网络结构信息的节点向量和边向量.

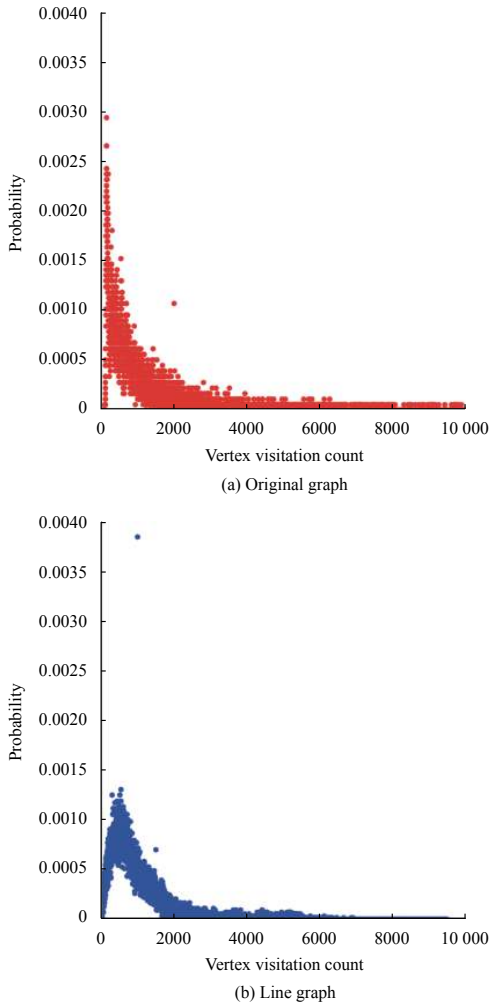


图 1 对 DBLP 数据集进行随机游走的节点采样频率分布

2.1 融合两种视角的网络表示学习

对于边视角, 本文首先根据原始网络构建边图 (line graph) 来获取边视角下的结构信息. 给定的图 $G(V,E)$, 其中V表示节点集合, E表示边集合. 令图G对应的边图为G_{link}. 在边图G_{link}中, 图 G 的边集合 E 映射为边图的节点集合V_{link}, 节点集合则映射为边图的边集合E_{link}. 通过这种转化构建的边图记为$G_{\text{link}} < V_{\text{link}}, E_{\text{link}} >$.

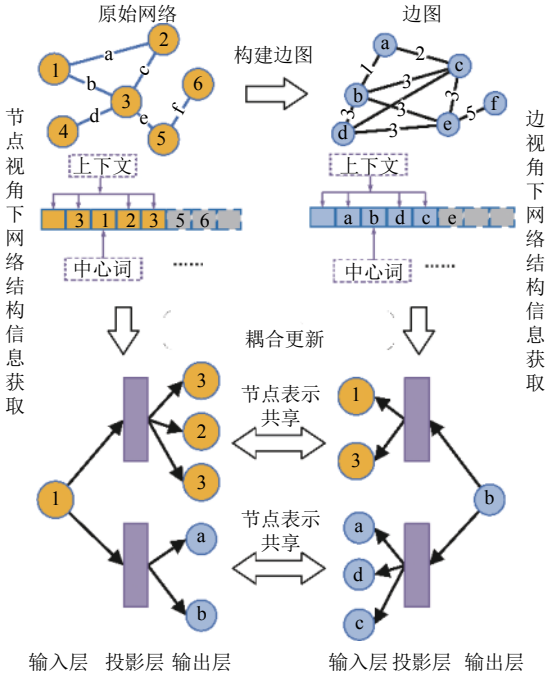


图 2 DPBCNE 模型框架示例

对于两种视角下信息的获取, 本文采用 DeepWalk 的方法. 通过对不同视角下的节点进行随机游走得到语料库, 每一个随机游走序列都可以看作是自然语言处理中的一个句子, 然后用 Skip-gram 算法最大化中心词节点和其上下文的共现概率, 从而得到不同视角下每个节点的表示向量, 使得在同一视角下具有相似结构的节点具有相似的向量.

因此, 双视角下网络结构信息获取的目标函数分别为:

$$L_1^{\text{node}} = \sum_{i=1}^{N} \sum_{s\in S} \sum_{-w\le j\le w, j\ne 0} \log_2 P(v_{i+j}|v_i) \quad (1)$$

$$L_1^{\text{link}} = \sum_{i=1}^{N^{\text{link}}} \sum_{s^{\text{link}}\in S^{\text{link}}} \sum_{-w\le j\le w, j\ne 0} \log_2 P\left(v_{i+j}^{\text{link}}|v_i^{\text{link}}\right) \quad (2)$$

其中, N为节点网络中节点的个数, S为随机游走得到的节点序列集合, w表示窗口大小, v表示节点网络中的节点. S^{link}为边图中随机游走得到的序列集合, N^{link}为边图中的节点个数即原始网络中边的数量, v^{link}表示边图中的节点也即原始网络中的边.

该部分总的目标函数为:

$$L_1 = L_1^{\text{node}} + L_1^{\text{link}} \tag{3}$$

为了得到有效的网络潜在特征, 本文同时融合节点视角和边视角进行网络表示学习. 介于节点与边的一对多的关系, 节点与其对应边的向量应当更相似, 通过不断耦合更新得到包含两种视角下网络拓扑信息的节点向量和边向量, 将另一视角的网络信息聚合到当前视角的节点向量学习上. 节点和边视角耦合关联模型目标函数分别如下:

$$L_{CO}^{\text{node}} = \sum_{i=1}^{N} \sum_{v_c^{\text{link}} \in C^{\text{node}}(v_i)} \log_2 P\left(v_c^{\text{link}} | v_i\right) \tag{4}$$

$$L_{CO}^{\text{link}} = \sum_{i=1}^{N} \sum_{v_c \in C^{\text{link}}\left(v_i^{\text{link}}\right)} \log_2 P\left(v_c | v_i^{\text{link}}\right) \tag{5}$$

关联部分总的目标函数为:

$$L_{CO} = L_{CO}^{\text{node}} + L_{CO}^{\text{link}} \tag{6}$$

其中, $C^{\text{node}}(v_i)$表示原始网络中与节点v_i直接相连的所有边的集合. $C^{\text{link}}\left(v_i^{\text{link}}\right)$表示原始网络中与边$v_i^{\text{link}}$直接相连的所有节点的集合.

根据上述内容, DBPBCNE 算法的总目标函数如下:

$$L = \alpha L_1 + (1-\alpha) L_{CO} \tag{7}$$

其中, $\alpha \in [0,1]$是用来平衡不同视角重要性的一个参数, 通过联合优化使得在节点学习的过程中耦合边视角下的信息, 以得到更有效的节点表示, 同时也能学习得到耦合两个视角下信息的边向量.

2.2 耦合训练过程

本文使用随机梯度上升 (SGA) 来训练模型, 通过 Softmax 公式来计算上述公式中的概率公式, 例如式 (1) 中使用的$P(v_{i+j}|v_i)$, 可以通过下列方式来计算:

$$P(v_{i+j}|v_i) = \frac{\exp\left(X'^{\mathrm{T}}_{v_{i+j}} \cdot X_{v_i}\right)}{\sum\limits_{v \in V, v \neq v_{i+j}} \exp\left(X'^{\mathrm{T}}_{v} \cdot X_{v_i}\right)} \tag{8}$$

为降低时间复杂度, 本文采用负采样的方法对公式 (8) 进行简化. 除了更新窗口中的已知邻居节点外, 为给定节点生成k个负样本. 在计算公式 (1) 中的梯度时, 便不需要在每个梯度步骤中枚举式 (8) 中的所有节点, 只需要计算根据词分布构建的负样本和已知的正样本. 假设给定邻居节点v_{i+j}的负样本集合为$NEG(v_{i+j}) = \{v_1, v_2, \cdots, v_k\}$则:

$$P(v_{i+j}|v_i) = \prod_{u \in \{v_{i+j}\} \cap NEG(v_{i+j})} P(u|v_i) \tag{9}$$

$$P(u|v_i) = \begin{cases} \sigma\left(X_{v_i}^{\mathrm{T}} \cdot X'_u\right), & I_1^{v_{i+j}}(u) = 1 \\ 1 - \sigma\left(X_{v_i}^{\mathrm{T}} \cdot X'_u\right), & I_1^{v_{i+j}}(u) = 0 \end{cases} \tag{10}$$

其中, X'_u表示待定参数, X_{v_i}为这部分要更新得到的节点向量, 指示函数$I_1^{v_{i+j}}(u) = \begin{cases} 1, & u = v_{i+j} \\ 0, & u \neq v_{i+j} \end{cases}$. 对$L_1^{\text{node}}$求导可以得出:

$$\frac{\partial L_1^{\text{node}}}{\partial X_{v_i}} = \sum\nolimits_{u \in NEG(v_{i+j})} \left[I_1^{v_{i+j}}(u) - \sigma\left(X_{v_i}^{\mathrm{T}} \cdot X'_u\right)\right] X'_u \tag{11}$$

$$\frac{\partial L_1^{\text{node}}}{\partial X'_u} = \left[I_1^{v_{i+j}}(u) - \sigma\left(X_{v_i}^{\mathrm{T}} \cdot X'_u\right)\right] X'_v \tag{12}$$

从而可以得到原始网络中的节点在节点视角下通过上下文节点进行更新的公式如下:

$$X'_u = X'_u + \eta\left[I_1^{v_{i+j}}(u) - \sigma\left(X_{v_i}^{\mathrm{T}} \cdot X'_u\right)\right] X_{v_i} \tag{13}$$

$$X_{v_i} = X_{v_i} + \eta \sum\nolimits_{u \in \{v_{i+j}\} \cap NEG(v_{i+j})} \left[I_1^{v_{i+j}}(u) - \sigma\left(X_{v_i}^{\mathrm{T}} \cdot X'_u\right)\right] X'_u \tag{14}$$

其中, η表示学习率. 同样的, 对于L_1^{link}也采用负采样以简化计算. 对于给定边视角下的已知的邻居节点v_{i+j}^{link}, 负样本集合为$NEG\left(v_i^{\text{link}}\right) = \left\{v_1^{\text{link}}, v_2^{\text{link}}, \cdots, v_k^{\text{link}}\right\}$, 边图中每个节点在边视角下通过上下文节点更新的公式如下:

$$X'_{u^{\text{link}}} = X'_{u^{\text{link}}} + \eta\left[I_1^{v_{i+j}^{\text{link}}}\left(u^{\text{link}}\right) - \sigma\left(X_{v_i^{\text{link}}}^{\mathrm{T}} \cdot X'_{u^{\text{link}}}\right)\right] X_{v_i^{\text{link}}} \tag{15}$$

$$X_{v_i^{\text{link}}} = X_{v_i^{\text{link}}} + \eta \sum\nolimits_{u \in \{v_{i+j}^{\text{link}}\} \cap NEG(v_{i+j}^{\text{link}})} \left[I_1^{v_{i+j}^{\text{link}}}\left(u^{\text{link}}\right) - \sigma\left(X_{v_i^{\text{link}}}^{\mathrm{T}} \cdot X'_{u^{\text{link}}}\right)\right] X'_{u^{\text{link}}} \tag{16}$$

其中, 指示函数为$I_1^{v_{i+j}^{\text{link}}}(u^{\text{link}}) = \begin{cases} 1, & u^{\text{link}} = v_{i+j}^{\text{link}} \\ 0, & u^{\text{link}} \neq v_{i+j}^{\text{link}} \end{cases}$.

对于节点-边耦合关联模型部分, 本文通过用原始网络中与边相连的节点更新边, 用与节点相连的边更新节点, 从而使得最终得到的原始网络的节点和边向量同时学习得到两个视角下的网络信息.

对于节点向量更新部分, 给定一个原始网络中的节点v_i和对应的一个相连边v_c^{link}, 其负采样得到的负样本集合为$NEG\left(v_c^{\text{link}}\right) = \left\{v_1^{\text{link}}, v_2^{\text{link}}, \cdots, v_k^{\text{link}}\right\}$, 同样, 对于边向量更新部分, 给定一个边图中的节点v_i^{link}和他对应的一个原始网络中的节点v_c, 其通过负采样得到的负样本集合为$NEG(v_c) = \{v_1, v_2, \cdots, v_k\}$

原始网络中节点在关联部分的向量更新公式如下:

$$X'_{u^{\text{link}}} = X'_{u^{\text{link}}} + \eta\left[I_{CO}^{v_c}\left(u^{\text{link}}\right) - \sigma\left(X_{v_i}^{T} \cdot X'_{u^{\text{link}}}\right)\right] X_{v_i} \quad (17)$$

$$X_{v_i} = X_{v_i} + \eta \sum_{u^{\text{link}} \in \{v^{\text{link}}\} \cap NEG(v^{\text{link}})} \left[I_{CO}^{v^{\text{link}}}\left(u^{\text{link}}\right) - \sigma\left(X_{v_i}^{\text{T}} \cdot X'_{u^{\text{link}}}\right)\right] X'_{u^{\text{link}}} \quad (18)$$

其中, 指示函数 $I_{CO}^{v^{\text{link}}}(u^{\text{link}}) = \begin{cases} 1, & u^{\text{link}} = v^{\text{link}} \\ 0, & u^{\text{link}} \neq v^{\text{link}} \end{cases}$.

原始网络中的边在关联部分的向量更新公式如下:

$$X'_u = X'_u + \eta\left[I_{CO}^{v_i^{\text{link}}}(u) - \sigma\left(X_{v_i^{\text{link}}}^{\text{T}} \cdot X'_u\right)\right] X_{v_i^{\text{link}}} \quad (19)$$

$$X_{v_i^{\text{link}}} = X_{v_i^{\text{link}}} + \eta \sum_{u \in \{v\} \cap NEG(v)} \left[I_{CO}^{v_c}(u) - \sigma\left(X_{v_i^{\text{link}}}^{T} \cdot X'_u\right)\right] X'_u \quad (20)$$

其中, 指示函数 $I_{CO}^{v_c}(u) = \begin{cases} 1, & u = v_c \\ 0, & u \neq v_c \end{cases}$.

为了更加详尽地介绍本文提出的 DPBCNE 算法的具体流程, 本文给出了算法伪代码如算法 1 所示.

算法 1. 基于双视角的耦合网络表示学习算法

输入: 图 $G(V,E)$, 窗口大小 w, 向量维度 d, 随机游走次数 γ, 总迭代次数 $ITER$, 游走长度 l, 学习率 η, 负采样个数 k.

输出: 节点表示向量矩阵 $X_v \in R^{N \times d}$, 边表示向量矩阵 $X_{v^{\text{link}}} \in R^{N^{\text{link}} \times d}$.

```
1. 初始化X, X^link
2. 根据G(V,E)构建边图G_link<V_link,E_link>
3.   For i=0 to γ:
4.     Ω=shuffle(V)
5.     Ω^link=shuffle(V^link)
6.     For each v_i∈Ω do
7.       S=RandomWalk(G,v_i,l)
8.     End for
9.     For each v_i^link∈Ω^link do
10.      S^link=RandomWalk(G^link,v_i^link,l)
11.    End for
12.  End for
13.  For iter=1,2,3,⋯,ITER do
14.    For each s∈S do
15.      根据式 (13) 和式 (14) 更新X_v和X'_v
16.      根据式 (15) 和式 (16) 更新X_v和X'_v^link
17.    End for
18.    For each s^link∈S^link do
19.      根据式 (18) 和式 (17) 更新X_v^link和X'_v^link
20.      根据式 (19) 和式 (20) 更新X_v^link和X'_v
21.    End for
22.  End for
```

算法可以分为第一步获取双视角下的网络结构信息部分和第二步耦合更新部分. 第 2–10 行分别对节点视角和边视角下的网络进行随机游走采样获取两个视角下的网络结构信息, 第 14–22 行表示耦合更新过程, 第 14–17 行对于第 1 步在原始网络中采样得到的所有节点序列中的每个节点, 分别用其对应的上下文节点和对应的边进行耦合更新, 同样, 第 18–21 行是对于边图中的采样得到的所有节点序列中的节点, 也即原始网络中的边, 分别用其上下文节点和对应的原始网络中的关联节点进行迭代耦合更新, 最终得到融合两个视角的原始网络的节点向量和边向量.

对于每一轮的迭代, 第一部分的对原始网络随机游走的时间复杂度为 Nrl, 对边图的随机游走时间复杂度为 $N^{\text{link}}rl$, 第二步耦合更新部分, 对于节点的更新过程时间复杂度为 $Ngdk + Ndwrlk$, 其中 g 表示原始网络中节点的平均度, 对于边的更新过程时间复杂度为 $2N^{\text{link}}dk + N^{\text{link}}dwrlk$, 其中, 每个网络的平均度通常都有一个最大值, 其余的 d, w, r, l, k 均为可以设置的常数.

3 实验分析

3.1 数据集

本文主要针对传统的网络表示学习研究只是通过节点视角这一单一的视角而没有考虑边视角的问题, 提出了一种新的基于双视角的耦合网络表示学习算法 (DPBCNE). 本文在 4 个真实的复杂网络数据集中分别验证了该方法的性能, 分别为 Facebook[21], GRQC[22], HEPTH[22] 和 DBLP[20] 数据集. 其中 Facebook 为社交网络, 总共有 193 种节点标签; GRQC 和 HEPTH 为合作者网络, 数据集中没有标签信息; DBLP 为引文网络, 总共有 4 种节点标签.

3.2 基准方法

在学习到节点的表示向量后, 本文分别在 4 个任务中与当下主流的网络表示学习算法进行比较, 其中, (1)–(3) 项是基于随机游走的网络表示学习算法, (4)–(6) 项是基于深度学习的网络表示学习算法, 此外, 针对链路预测和链路重构任务, 本文还选择了一种效果良好的传统链路预测方法 Common Neighbor[23] 作为对比方法, 相关基准算法简介如下:

(1) DeepWalk[8]: 对节点采用随机游走和 Skip-gram 模型以学习得到每个节点的表示向量.

(2) Node2Vec[9]: 在 DeepWalk 的基础上, 引入了带偏置的随机游走, 以选择不同的搜索方式采样节点. 其中, 偏置参数 $p = 0.25$, $q = 0.5$.

(3) Line[11]: 通过分别定义损失函数同时保存网络中节点的一阶邻近度和二阶邻近度.

(4) SDNE[12]: 使用自动编码器通过联合优化目标函数来保持节点一阶和二阶邻近性. 该方法采用高度非线性的函数对网络的邻接矩阵进行编码.

(5) DNGR[13]: 利用 Random Surfing 策略生成概率共现矩阵, 再作为叠加去噪自动编码器的输入进行节点表示的学习.

(6) GAE[14]: 使用图卷积网络 (GCN) 编码器和内积解码器. 该方法利用 GCN 学习节点间的高阶关系.

(7) Common Neighbor (CN)[20]: 以每个节点对之间的公共邻居数作为节点之间的相似度评分, 以进行链路预测.

在实验过程中, 为保证公平比较, 所有实验的参数均统一设置. 对于网络表示学习算法, 其维度为 128, 负采样样本数为 5, 窗口大小为 5, 每个节点的随机游走次数为 10, 步长是 40, 边图的设置与原始网络相同. 学习率为 0.01, 最大迭代次数为 200.

3.3 评价指标

为验证 DPBCNE 的有效性, 本文分别在链路预测, 链路重构, 节点分类和边分类 4 个任务上进行了对比实验.

对于链路预测和链路重构, 采用了 *AUC* (Area Under Curve) 和 *AP* (Average Precision) 指标来验证最终的效果.

AUC 表示当随机选择一个正样本和一个负样本时, 正样本分数高于负样本的概率. 例如, 在链路预测任务中, 随机挑选测试集中的一条边和一条不存在的边并进行比较, 重复进行 n 次, 其中有n'次测试集中边的分数大于不存在的边, 有n''次两者获得同样的分数, 那么最终得到的 *AUC* 计算公式如下:

$$AUC = \frac{n' + 0.5n''}{n} \tag{21}$$

AP 表示平均准确率, 其计算公式如下:

$$AP = \frac{\sum Precision}{M} \tag{22}$$

其中, $Precision$ 表示每个类别的准确率, M 表示类别数.

对于节点分类和边分类实验, 采用了 *Micro-F*1 和 *Macro-F*1 来作为评价指标. 定义如下:

$$Micro - F1 = \frac{\sum_{A \in C} F1(A)}{|C|} \tag{23}$$

$$Macro - F1 = \frac{2 \cdot Pr \cdot R}{Pr + R} \tag{24}$$

其中, $F1(A)$ 表示标签 A 的 $F1$ 得分, C 表示所有的标签集. Pr 表示总的准确率, R 表示总的召回率.

3.4 节点分类

节点分类是网络表示学习中用以验证算法有效性的一个重要任务. 此任务在 Facebook 和 DBLP 数据集上验证了 DPBCNE 算法, 首先移除数据集中没有标签的节点, 将数据集按照 30% 的比例划分训练集, 剩余的节点作为测试集, 将每个节点学习得到的表示向量作为逻辑回归分类器的输入进行训练, 通过计算 *Micro-F*1 和 *Macro-F*1 来比较不同模型之间的效果, 最终结果如表 1 所示.

表 1 节点分类实验结果

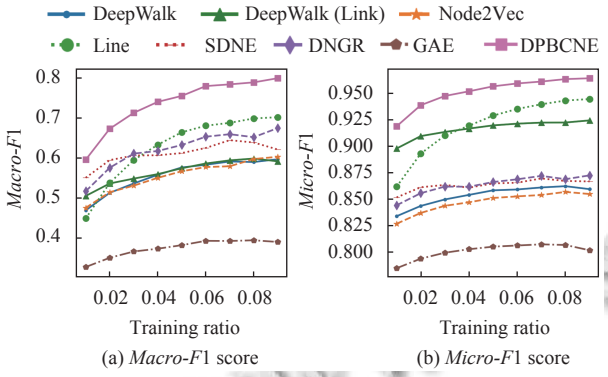
算法	Facebook		DBLP	
	*Macro-F*1	*Micro-F*1	*Macro-F*1	*Micro-F*1
DeepWalk	0.3060	0.7283	0.6702	0.7483
Node2Vec	0.3012	0.7273	0.6685	0.7465
Line	0.2656	0.7215	0.4776	0.6219
SDNE	0.2976	0.6645	0.3457	0.5471
DNGR	0.3190	0.7443	0.7292	0.7904
GAE	0.2325	0.6926	0.6714	0.7513
DPBCNE	**0.3289**	**0.7562**	**0.7347**	**0.7934**

从表 1 中可以看到, DPBCNE 模型地结果在两个数据集上都优于其他算法. 在 Facebook 数据集中, DPBCNE 的 *Macro-F*1 得分比其他算法中表现最好的 DNGR 算法高出了 0.99%, *Micro-F*1 得分则比 DNGR 算法高出了 1.19%. 这表明, 通过融合两种视角下的网络表示学习能够获取比在单一视角下更丰富的节点采样结果, 也就是更丰富的网络结构信息, 使得其在节点的类别划分上比只关注单一视角的效果更好. 而在 DBLP 数据集中, DPBCNE 算法的 *Macro-F*1 和 *Micro-F*1 得分比 DNGR 分别高了 0.3% 和 0.55%. DPBCNE 模型在 Facebook 数据集中的提升效果比在 DBLP 数据集中更好, 这是因为 Facebook 数据集中每个节点具有多个标签, 而 DBLP 数据集的每个节点有且只有一个标签, 融合了边视角的耦合网络表示学习, 也更能区分出节点的重叠社区.

3.5 边分类

传统的网络表示学习算法通常使用两个节点向量简单相连或相加来作为两个节点之间的边的表示[9]. 在本任务中, 对于所有的网络表示学习基准算法, 首先学习到每个节点的向量, 再用$(v_i + v_j)/2$表示节点v_i和节点v_j之间的边向量. 而对于 DPBCNE, 通过耦合学习, 可

以直接得到每条边的表示, 使其更具解释性. 本任务采用 Facebook 数据集进行验证, 聚合每对节点的标签, 作为其对应边的标签, 按照 1%~9% 的比例对于网络中的边划分出训练集作为逻辑回归分类器的输入. 本任务采用了 *Micro-F*1 和 *Macro-F*1 指标来衡量最终的效果, 具体结果如图 3 所示.



(a) *Macro-F*1 score (b) *Micro-F*1 score

图 3 边分类实验结果

图 3 中, DeepWalk (Link) 方法通过直接对边图的节点进行随机游走, 不进行耦合训练直接得到边向量. DeepWalk (Link) 在边分类任务中也取得了比只在节点上进行 DeepWalk 学习节点表示再拼接成边向量更好的结果, 这说明对于边向量的计算, 在边图上直接进行学习得到是有效的. 同时, DPBCNE 始终高于其他所有方法包括效果良好的 DeepWalk (Link), 这说明融合两个视角的耦合训练学习得到的边向量可以更好地保存网络中的信息.

3.6 链路预测

在这个任务中, 首先将网络中所有的边划分为测试集和训练集 (比例为 3:7), 同时保持网络的连通性, 通过对训练集进行网络表示学习, 得到网络中每个节点的表示, 再计算$|v_i - v_j|$作为v_i和v_j之间的边的表示. 本文认为两个节点之间是否存在边可以由节点对应向量的绝对差来决定[4]. 将测试集中的边看作为正例, 对每一个正例等比例地构建一条不存在的边当作负例, 将其作为逻辑回归分类器的输入. 本任务用 *AUC* 和 *AP* 作为本任务的评价指标, 具体结果如表 2 所示.

由表 2 可以看到, DPBCNE 的 *AUC* 指标都在 75% 以上, 这说明 DPBCNE 可以有效地预测网络中的未知边. 在 Facebook 数据集以及 DBLP 数据集中, DPBCNE 取得了最好的效果, 在 GRQC 与 HEPTH 数据集中 DPBCNE 的效果仅次于 CN 算法, 这说明在合作者网络中, 节点之间的关系非常受共同邻居的影响, 复杂的学习反而没有简单的指标来得有效, 但 DPBCNE 仍然高于其他所有基于随机游走和深度学习的网络表示学习, 由于融合了两种视角, DPBCNE 能够更好地预测网络中未知的边.

表 2 链路预测实验结果

算法	Facebook		GRQC		HEPTH		DBLP	
	AUC	*AP*	*AUC*	*AP*	*AUC*	*AP*	*AUC*	*AP*
CN	0.9275	0.9164	**0.9162**	**0.9127**	**0.8652**	**0.8629**	0.7494	0.7478
DeepWalk	0.8930	0.8568	0.6707	0.6155	0.5811	0.5470	0.5916	0.5549
Node2Vec	0.9048	0.8721	0.6831	0.6264	0.5784	0.5452	0.5962	0.5578
Line	0.9284	0.8984	0.7201	0.6597	0.5989	0.5596	0.6042	0.5639
SDNE	0.7376	0.6886	0.6268	0.5821	0.5405	0.5233	0.5886	0.5534
DNGR	0.9352	0.9080	0.8458	0.7971	0.8055	0.7512	0.7219	0.7011
GAE	0.9073	0.8765	0.8088	0.7601	0.7230	0.6673	0.6908	0.6330
DPBCNE	**0.9398**	**0.9092**	0.9092	0.8838	0.8185	0.7737	**0.8135**	**0.7594**

3.7 链路重构

链路重构任务类似于链路预测, 不同的是链路重构所重构的是现有的边, 而不是去预测未知的边. 给定一个网络, 使用不同的链路重构方法来重构原始网络的所有边. 在这个任务中, 依旧使用两个节点表示向量之间的绝对差作为每条边的表示. 同样, 采用 *AUC* 和 *AP* 作为评价指标. 具体结果如表 3 所示.

可以看到, DPBCNE 模型的 AUC 结果都接近于 1, 这说明该方法能够很好地保存网络的邻接关系. 在 4 个数据集中, DPBCNE 始终效果是最好的, 相比于传统的链路预测方法 CN, DPBCNE 在 *AUC* 指标上提升了 1.03%~17.59%, 在 *AP* 指标上提升了 0.87%~17.16%, 相比于效果最好的基于随机游走的算法 LINE, DPBCNE 在 *AUC* 指标上提升了 0.96%~19.05%, 在 *AP* 指标上提升了 0.69%~24.1%, 对比效果最好的基于深度学习的算法 DNGR, DPBCNE 在 *AUC* 指标上提升了 1.76%~

5.3%, 在 *AP* 指标上提升了 1.7%~7.73%.

4 结论与展望

本文讨论了两种不同视角在社区结构和随机游走方面的差异, 通过经验分析, 本文得出边视角和节点视角在随机游走中出现的节点分布是不同的, 这意味着通过不同的视角, 可以获得更多的网络拓扑信息. 因此, 本文提出了一种新的网络表示学习算法 DPBCNE, 可以同时考虑边视角和节点视角, 并通过耦合学习, 学习得到节点向量和边向量. 本文在节点分类, 边分类, 链路预测和链路重构 4 个任务上验证了该方法的效果. DPBCNE 在 4 个任务中都展现了其良好的性能. 本文只在静态网络中进行了任务验证, 而在现实生活中, 网络是不断变化的, 因此, 未来的研究 将该算法考虑扩展到动态网络以及加入深度学习框架以获得更好的效果[24]. 同时, 鉴于 DPBCNE 可以直接学习得到每条边的表示, 还将考虑将其扩展到知识图谱的应用研究中[25].

表 3 链路重构实验结果

算法	Facebook		GRQC		HEPTH		DBLP	
	AUC	*AP*	*AUC*	*AP*	*AUC*	*AP*	*AUC*	*AP*
CN	0.9314	0.9203	0.9387	0.9331	0.9264	0.9215	0.8059	0.8001
DeepWalk	0.9079	0.8762	0.7441	0.6815	0.6726	0.6172	0.6845	0.6283
Node2Vec	0.9171	0.8898	0.7639	0.7033	0.6710	0.6154	0.6959	0.6385
Line	0.9441	0.9221	0.8579	0.8087	0.8069	0.7500	0.7913	0.7306
SDNE	0.7579	0.7080	0.6851	0.6305	0.5837	0.5490	0.5754	0.5441
DNGR	0.9361	0.9120	0.9260	0.8940	0.9255	0.8948	0.9313	0.8981
GAE	0.9114	0.8823	0.8238	0.9015	0.8893	0.8556	0.8286	0.7774
DPBCNE	**0.9537**	**0.9290**	**0.9790**	**0.9713**	**0.9781**	**0.9678**	**0.9818**	**0.9716**

参考文献

1 安沈昊, 于荣欢. 复杂网络理论研究综述. 计算机系统应用, 2020, 29(9): 26–31. [doi: 10.15888/j.cnki.csa.007617]

2 吕琳媛. 复杂网络链路预测. 电子科技大学学报, 2010, 39(5): 651–661. [doi: 10.3969/j.issn.1001-0548.2010.05.002]

3 郝志峰, 柯妍蓉, 李烁, 等. 基于图编码网络的社交网络节点分类方法. 计算机应用, 2020, 40(1): 188–195.

4 Zhou LK, Yang Y, Ren X, *et al*. Dynamic network embedding by modeling triadic closure process. Proceedings of the 32nd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. New Orleans, LA, USA. 2018. 571–578.

5 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.

6 冶忠林, 赵海兴, 张科, 等. 基于多视图集成的网络表示学习算法. 计算机科学, 2019, 46(1): 117–125. [doi: 10.11896/j.issn.1002-137X.2019.01.018]

7 王杰, 张曦煌. 基于图卷积网络和自编码器的半监督网络表示学习模型. 模式识别与人工智能, 2019, 32(4): 317–325.

8 Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. East Lansing, MI, USA. 2014. 701–710.

9 Grover A, Leskovec J. node2vec: Scalable feature learning for networks. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. East Lansing, MI, USA. 2016. 855–864.

10 Mikolov T, Chen K, Conrado G, *et al*. Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR. Scottsdale, AZ, USA. 2013. 1–12.

11 Tang J, Qu M, Wang MZ, *et al*. LINE: Large-scale information network embedding. Proceedings of the 24th International Conference on World Wide Web. Florence, Italy. 2015. 1067–1077.

12 Wang DX, Cui P, Zhu WW. Structural deep network embedding. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. East Lansing, MI, USA. 2016. 1225–1234.

13 Cao SS, Lu W, Xu QK. Deep neural networks for learning graph representations. Proceedings of the 30th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. Phoenix, AZ, USA. 2016. 1145–1152.

14 Kipf TN, Welling M. Variational graph auto-encoders. Proceedings of NIPS Workshop on Bayesian Deep Learning. Cambridge, UK. 2016.

15 陈丽, 朱裴松, 钱铁云, 等. 基于边采样的网络表示学习模型. 软件学报, 2018, 29(3): 756–771. [doi: 10.13328/j.cnki.jos.005435]

16 Goyal P, Hosseinmardi H, Ferrara E, *et al*. Capturing edge attributes via network embedding. IEEE Transactions on

Computational Social Systems, 2018, 5(4): 907–917. [doi: 10.1109/TCSS.2018.2877083]

17 Whitney H. Congruent graphs and the connectivity of graphs. In: Eells J, Toledo D, eds. Hassler Whitney Collected Papers. Boston: Birkhäuser, 1992. 61–79.

18 Ahn YY, Bagrow JP, Lehmann S. Link communities reveal multiscale complexity in networks. Nature, 2010, 466(7307): 761–764. [doi: 10.1038/nature09182]

19 金弟, 杨博, 刘杰, 等. 复杂网络簇结构探测——基于随机游走的蚁群算法. 软件学报, 2012, 23(3): 451–464. [doi: 10.3724/SP.J.1001.2012.03996]

20 Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): 2. [doi: 10.1145/1217299.1217301]

21 McAuley J, Leskovec J. Learning to discover social circles in ego networks. Advances in Neural Information Processing Systems 25. Lake Tahoe, NV, USA. 2012. 539–547.

22 Tang J, Zhang J, Yao LM, *et al*. ArnetMiner: Extraction and mining of academic social networks. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas, NV, USA. 2008. 990–998.

23 Liben-Nowell D, Kleinberg J. The link prediction problem for social networks. Proceedings of the Twelfth International Conference on Information and Knowledge Management. New Orleans, LA, USA. 2003. 556–559.

24 崔广新, 李殿奎. 基于自编码算法的深度学习综述. 计算机系统应用, 2018, 27(9): 47–51. [doi: 10.15888/j.cnki.csa.006542]

25 黄恒琪, 于娟, 廖晓, 等. 知识图谱研究综述. 计算机系统应用, 2019, 28(6): 1–12. [doi: 10.15888/j.cnki.csa.006915]