·移动互联与通信技术· 文章编号：1000-3428(2021)06-0188-09 文献标志码：A 中图分类号：TP181

基于信息传播节点集的CTDN节点分类算法

黄 鑫[1,2]，李 赟[2]，熊瑾煜[2]

（1.中国人民解放军战略支援部队信息工程大学 信息系统工程学院，郑州 450001；2.盲信号处理国家级重点实验室，成都 610041）

摘 要：针对连续时间动态网络的节点分类问题，根据实际网络信息传播特点定义信息传播节点集，改进网络表示学习的节点序列采样策略，并设计基于信息传播节点集的连续时间动态网络节点分类算法，通过网络表示学习方法生成的节点低维向量以及OpenNE框架内的LogicRegression分类器，获得连续时间动态网络的节点分类结果。实验结果表明，与CTDNE和STWalk算法相比，该算法在实验条件相同的情况下，网络表示学习结果的二维可视化效果更优且最终的网络节点分类精度更高。

关键词：信息传播节点集；连续时间动态网络；网络表示学习；节点分类；随机游走；Skip-Gram模型

开放科学（资源服务）标志码（OSID）：

中文引用格式：黄鑫，李赟，熊瑾煜.基于信息传播节点集的CTDN节点分类算法[J].计算机工程，2021，47(6)：188-196.

英文引用格式：HUANG Xin，LI Yun，XIONG Jinyu.Node classification algorithm based on information propagation node set for CTDN[J].Computer Engineering，2021，47(6)：188-196.

Node Classification Algorithm Based on Information Propagation Node Set for CTDN

HUANG Xin[1,2]，LI Yun[2]，XIONG Jinyu[2]

（1.College of Information System Engineering，PLA Strategic Support Force Information Engineering University，Zhengzhou 450001，China；2.National Key Laboratory of Science and Technology on Blind Signal Processing，Chengdu 610041，China）

【Abstract】The study described in this paper addresses the problem of node classification in Continuous-Time Dynamic Network（CTDN）.In this work，an information propagation node set is defined according to the features of the actual network information propagation，and the node sequence sampling strategy in network representation learning is improved.Based on the defined information propagation node set，a node classification algorithm for CTDN is designed.The algorithm employs the network representation method to generate the low-dimensional node vector，and uses the LogicRegression classifier to obtain the node classification results of CTDN.Experimental results show that the proposed algorithm outperforms the existing classic algorithms such as CTDNE and STWalk under the same experimental conditions，providing better 2D visualized network representation learning results and higher network node classification accuracy.

【Key words】information propagation node set；Continuous-Time Dynamic Network（CTDN）；network representation learning；node classification；random walk；Skip-Gram model

DOI：10. 19678/j. issn. 1000-3428. 0058719

0 概述

复杂网络是对现实世界中复杂系统的抽象表示，复杂系统的各组成部分及其相互之间的关联关系分别用节点和节点之间的边来表示。对复杂网络中的节点开展分类问题研究，有利于加深对复杂系统内部组成的理解。传统网络节点分类主要针对静态网络，即不考虑网络随时间发生演变，网络节点和节点之间的边始终保持不变。在实际情况中，网络的动态特征明显，节点和节点之间的边可能随时间发生变化。为使研究更贴近实际情况，在静态网络的基础上充分考虑时间要素，研究人员提出动态网络概念并进一步开展网络相关研究。本文基于经典的网络节点分类方法，在考虑时间要素的前提下，根据连续时间动态网络（Continuous-Time Dynamic Network，CTDN）的信息传播特征，结合网络表示学习方法进行网络节点分类研

基金项目：国防科技重点实验室基金。

作者简介：黄 鑫（1988—），男，工程师、硕士研究生，主研方向为智能信息处理；李 赟，助理研究员、博士；熊瑾煜，副研究员、博士。

收稿日期：2020-06-22 修回日期：2020-08-11 E-mail：Huangx_0735@163.com

究,提出基于信息传播节点集的连续时间动态网络节点分类算法CTDNN-IPNS。

1 相关工作

基于网络表示学习的节点分类方法是研究网络节点分类问题的一类重要方法[1-3]。这类方法将网络节点表示为低维空间向量,通过对向量的分类实现节点的分类。结合网络表示学习方法,在分类过程中根据是否考虑节点或连边随时间变化的情况,形成静态网络节点分类和动态网络节点分类方法。

根据网络表示学习模型的使用情况,静态网络节点分类方法[4]大致可分为基于矩阵分解[5-6]、随机游走[7-8]和深度神经网络[9-10]三类。基于随机游走的网络表示学习方法将随机游走与自然语言处理领域的Skip-Gram词向量生成模型相结合,形成节点采样+Skip-Gram的网络表示学习框架,将网络节点和通过在网络节点之间随机游走采样获取的节点序列分别视作自然语言中的词和语句,对节点序列加以处理后实现网络节点的向量表示,并利用经典分类算法实现网络节点的最终分类。DeepWalk[7]算法是经典的基于随机游走的网络表示学习算法,具有网络表示能力强和计算复杂度低的特点。在此基础上,通过改进节点序列采样策略,衍生出Node2Vec[8]、Walklets[11]、Metapath2Vec[12]等众多网络表示学习算法。这类算法针对静态网络展开研究,未能对网络中的时间信息加以利用,即未考虑节点或连边随时间的变化情况对网络表示学习结果的影响,不适用于动态网络节点分类。

在动态网络节点分类方面,文献[13-14]利用LSTM、AutoEncoder等深度学习模型对网络快照进行处理,较好地表示出网络节点类别随时间的演化过程,但是如果节点在不同的快照中表现出不同的类别,则这类方法不能给出节点的全局类别属性。文献[15-17]以改进节点序列采样策略为突破口,分别设计出基于随机游走的动态网络表示学习算法CTDNE、STWalk和RWR-STNE,其中,STWalk和RWR-STNE算法在静态网络的基础上增加时间要素,在不同时刻网络快照上构造节点时空图,进而在其上完成随机游走并实现节点采样,但是上述算法存在时间粒度过大、时间信息利用不充分的问题。CTDNE算法针对连续时间动态网络,严格依照事件发生的时间顺序进行节点采样,但容易受噪声影响,导致网络表示学习结果与现实情况存在较大偏差,分类结果精度也会随之降低。

2 相关定义

定义1(连续时间动态网络) 连续时间动态网络[15,18-19]表示为图$G=(V,E_T,T)$,其中,V为节点集,$E_T\subseteq V\times V\times\mathbb{R}^+$为任意两个节点间具有时间戳的连边集,$T:E_T\to\mathbb{R}^+$为边的时间戳值到正实数集的映射。$e_i=(u,v,t)\in E_T$表示网络中的边,其中,$u$为源节点,$v$为目的节点,$t$为连边发生的时间戳。在最小时间粒度情况下,每条边可能具有互不相同的时间戳值。

连续时间动态网络的定义在传统静态网络的基础上充分考虑了动态网络中边的时序信息,同时克服了以网络快照形式表示动态网络过程中时间信息损失的问题。针对连续时间动态网络进行节点分类的3个主要步骤如图1所示。首先,按照定义1,利用实际数据构造连续时间动态网络;其次,使用网络表示学习方法,将连续时间动态网络中的节点映射至低维空间,采用保有节点原始关系的向量加以表示;最后,利用分类算法,通过对低维空间节点向量的分类,实现连续时间动态网络的节点分类。鉴于分类算法已经相对成熟,本文将网络表示学习环节作为研究重点,开展连续时间动态网络节点分类研究。

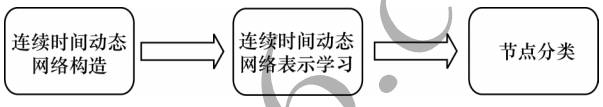


图1 连续时间动态网络节点分类流程

Fig.1 Procedure of node classification for CTDN

定义2(连续时间动态网络表示学习) 在连续时间动态网络中,学习得到的映射函数$f:V\to\mathbb{R}^d$,使得网络中的节点$v_i\in V$被映射为低维向量$\boldsymbol{m}_i\in\mathbb{R}^d$,其中,$d$表示向量维度且满足$d<<|V|$。

在通常情况下,映射函数f的目标是保留节点在原始网络结构上的内在相似性和时间上的平滑性。

3 CTDNN-IPNS算法

在网络信息传播动力学研究中,DALEY等人[20]提出了经典的DK谣言传播模型。该模型将网络内节点分为与谣言传播无关者、传播谣言者和知道谣言但不继续传播者、谣言通过传播者之间的直接接触进行传播三类。在谣言传播过程中,节点间因接触范围不同,形成谣言传播群组,群组内因节点传播能力的不同,会产生不同的传播模式。

在实际网络信息传播过程中,信息通过节点间通联在不同类型节点之间传播。因此,在DK谣言传播模型的基础上,本文结合实际通信网络数据特点及其时间维度属性,得出连续时间动态网络具有以下特点:

1)信息传播流程多数在一定时间内完成,传播范围在大小不等的节点集内。

2)信息传播包括一对一、一对多和多对多等多种模式。

3)类别相同或相似节点之间存在一定的周期性关联关系。

电话通信网络是典型的连续时间动态网络。表1是某电话通信网络的部分通话记录,在时间戳值为316 999 s~317 344 s的345 s时间内,其中方括号标注的用户171、180、186、188共同完成一次信息传播,而其他用户与其没有任何通联。图2为信息传播过程示例,其中连边上的数字表示通联发生的时间顺序。

表1 某电话通信网络部分通话记录

Table 1 Partial call records of a telephone communication network

源节点编号	目的节点编号	时间戳值/s	源节点编号	目的节点编号	时间戳值/s
4	288	316 999	191	193	317 170
[180]	[171]	317 017	118	128	317 174
245	262	317 073	194	196	317 233
[186]	[171]	317 086	177	176	317 261
194	192	317 108	[180]	[171]	317 283
240	266	317 125	200	197	317 288
65	61	317 146	[180]	[188]	317 324
[180]	[188]	317 149	193	191	317 335
114	111	317 165	314	107	317 344

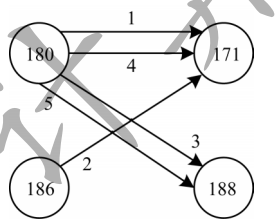


图2 用户171、180、186和188之间的信息传播过程

Fig.2 Information dissemination process among users 171, 180, 186 and 188

由此可以推测,对于连续时间动态网络的节点分类,若在网络表示学习环节的节点序列采样过程中对上述特征加以利用,其网络表示学习结果将更好地保留节点在原始网络结构上的内在相似性,在此基础上得到的分类结果精度也将大幅提高。具体而言:一是将节点采样范围、时间范围加以限制,提高节点集内成员共现概率;二是增加采样过程的灵活性,从逐个节点顺序采样转变为从节点集内成员发起的随机采样;三是信息传播周期性的存在,使得同类节点共现概率会在一定范围内随采样次数的提高而增加。

定义3(信息传播节点集) 给定连续时间动态网络G,在时间范围Δt内,在信息I从节点v_i传播至节点v_j的过程中,所有参与此次信息传播的节点记为$M=\{v_i,\cdots,v_k,\cdots,v_j\}$,$v_p\in V$,$p\in\{i,\cdots,k,\cdots,j\}$,这些节点共同组成信息$I$的传播节点集。

基于上述分析,本文提出针对连续时间动态网络节点分类的CTDNN-IPNS算法,该算法基于信息传播节点集的概念,对网络表示学习环节的节点序列采样策略进行改进,形成突显节点之间关联关系的节点向量表示,在此基础上进行类别划分,最终实现对连续时间动态网络节点的分类。

3.1 节点序列采样

节点序列采样的具体步骤如下:

步骤1 构造连续时间动态网络$G=(V,E_T,T)$,分别初始化信息传播节点集M、备选边集E_c和节点采样序列L为$\varnothing$,设置信息传播时间范围Δt、节点序列长度(即随机游走步长)l及采样次数(即随机游走次数)n。

步骤2 从G中随机选择一条边作为初始边,其时间戳t作为本轮采样的基准时间,而其两端节点则作为初始节点加入信息传播节点集M。

步骤3 与M中节点相连的所有边,若其时间戳在时间$t\pm\Delta t$内,则将其置入备选边集E_c。

步骤4 若$E_c\neq\varnothing$,则从E_c中随机选择一条边作为下一步采样的起始边,之后操作与步骤2类似,但需合并M中的相同节点,并将新增节点添加至L。

步骤5 若$E_c=\varnothing$,则在时间$t+\Delta t$内随机调整基准时间t,重复步骤3~步骤5。

步骤6 重复步骤2~步骤5,当$|M|\geqslant l$或$t\pm\Delta t$超出G的时间范围时,输出节点采样序列L。

步骤7 重复步骤2~步骤6共c次,输出n个节点序列$L_1,L_2,\cdots,L_n$。

算法1 CTDNN-IPNS算法

输入 连续时间动态网络$G=(V,E_T,T)$、时间范围Δt、随机游走步长l、随机游走次数n

输出 节点序列L_{list}

1.初始化信息传播节点集M、备选边集E_c为∅,设置采样基准时间t = 0

2.for 1 to n

3.随机选择起始边e_s,$e_s\in E_T$,设置t为e_s的时间戳

4.while|M| < l

5.将e_s的两端节点添加至M,合并其中的相同节点

6.for 与M中节点相连的所有边

7.获取边的时间戳

8.if t − Δt ≤ 边的时间戳 ≤ t + Δt

9.将该边添加至E_c

10.if $E_c\neq\varnothing$

11.从E_c中随机选择一条边e,令e_s = e,更新t为e的时间戳

12.else

13.在t ± Δt范围内随机调整t值

14.if t ± Δt超出G的时间范围
15.退出本次while循环
16.将M添加至L_{list}
17.return L_{list}

在算法1中,输入参数l控制每次节点序列采样的最大长度,n表示最终形成的节点序列个数,Δt表示一次信息传播的时间范围。

3.2 网络表示学习

定义4(节点邻居序列) 对于网络中的节点u,在以采样策略S进行一次采样形成的序列中,与其同时被采集到的节点构成节点u的节点邻居序列[7],记为$N_S(v)\subset V$。

基于节点采样+Skip-Gram的网络表示学习框架,可将网络表示学习问题转化为使V中所有节点$v\in V$在嵌入结果为$f(v)$的条件下,节点邻居序列中的节点共同出现的对数条件概率之和最大的优化问题,计算公式为:

$$\max_f \ln P\left(N_S(v)\middle|f(v)\right) \tag{1}$$

为简化计算过程进行以下假设:

1)假设不同节点之间的采样过程相互独立,则如式(1)所示的条件概率可表示为$N_S(v)$内各节点的条件概率之积,计算公式为:

$$P\left(N_S(v)\middle|f(v)\right)=P\left(n_i\middle|f(v)\right) \tag{2}$$

2)假设同一条边的两端节点彼此作用对称,利用softmax函数表示式(2)中的条件概率,计算公式为:

$$P\left(n_i\middle|f(v)\right)=\frac{\exp(f(n_i)\cdot f(v))}{\sum\limits_{u\in V}\exp(f(u)\cdot f(v))} \tag{3}$$

基于上述假设,式(1)可简化为:

$$\max_f \sum_{v\in V}\left[-\ln Z_v+\sum_{n_i\in N_S(v)} f(n_i)f(v)\right] \tag{4}$$

其中,$Z_v=\sum\limits_{u\in V}\exp(f(v)f(u))$。

由上述公式可知,网络表示学习的目标函数求解的关键为构造$N_s(v)$,即采样策略S的设计。利用节点采样+Skip-Gram的网络表示学习框架,通过负采样方法[21]和Skip-Gram模型即可求解式(4)描述的目标函数,从而生成网络节点的d维向量表示,其中d值由人为设定。需要说明的是,若要生成网络节点d维向量表示,则需从节点序列中截取其子序列作为Skip-Gram模型输入,而截取考察范围w同样由人为设定,该参数表示在截取节点序列的子序列时,针对节点v_i截取的节点子序列为$\{v_{i-w},\cdots,v_i,\cdots,v_{i+w}\}$。

3.3 节点向量分类

CTDNN-IPNS算法采用LogicRegression分类器对网络表示学习环节生成的节点向量进行分类,并依据F1_macro和F1_micro值量化评价分类结果。F1_macro和F1_micro的求解过程为:设数据集中的数据共分为n类,类别集合为$C=\{c_1,c_2,\cdots,c_n\}$。对于类别c_i,$i=1,2,\cdots,n$,数据分类结果中的正确分类样本、错误分类样本和非c_i类错误分类样本数量可分别表示为T_{TP}、F_{FP}、F_{FN},则F1_macro可根据式(5)~式(8)进行求解,反映了分类结果在各个类别中样本分类的综合性能,F1_micro可根据式(9)~式(11)进行求解,反映了分类结果在所有样本上的综合分类性能。

$$P_{\text{Precision}}=\frac{T_{TP}}{T_{TP}+F_{FP}} \tag{5}$$

$$R_{\text{Recall}}=\frac{T_{TP}}{T_{TP}+F_{FN}} \tag{6}$$

$$F=\frac{2\cdot P_{\text{Precision}}\cdot R_{\text{Recall}}}{P_{\text{Precision}}+R_{\text{Recall}}} \tag{7}$$

$$F_{\text{F1\_macro}}=\frac{\sum\limits_{c_i\in C}F}{|C|} \tag{8}$$

$$P_{\text{Precision}}=\frac{\sum\limits_{c_i\in C}T_{TP}}{\sum\limits_{c_i\in C}(T_{TP}+F_{FP})} \tag{9}$$

$$R_{\text{Recall}}=\frac{\sum\limits_{c_i\in C}T_{TP}}{\sum\limits_{c_i\in C}(T_{TP}+F_{FN})} \tag{10}$$

$$F_{\text{F1\_micro}}=\frac{2\cdot P_{\text{Precision}}\cdot R_{\text{Recall}}}{P_{\text{Precision}}+R_{\text{Recall}}} \tag{11}$$

4 实验与结果分析

4.1 实验数据集与环境

本文选用网络表示学习研究领域常用的DBLP和AMiner论文合作数据集,以及根据实际电话通联记录自制的Reality-Call数据集,从连续时间动态网络的二维可视化展示效果及其节点分类结果两方面,对CTDNN-IPNS算法的性能进行实验验证,数据集信息如表2所示。DBLP和AMiner数据集中的网络节点是文章作者,若两位作者共同发表过论文,则两者之间存在一条连边,边的时间戳为论文发表年份,节点类别是论文作者的所属研究领域。类似地,Reality-Call数据集中的用户号码被视为网络节点,若两位用户有过通话,则其对应的节点之间存在一条连边,边的时间戳为通话发起时间,节点类别为号码所属部门。实验环境设置如表3所示。

表2 实验数据集

Table 2 Experimental dataset

数据集	节点数	边数	类别数	时间范围	网络平均路径长度
DBLP	31 572	123 615	6	2000年—2010年	5.4
AMiner	10 988	34 831	5	2000年—2015年	9.9
Reality-Call	214	197 115	4	24天	2.9

表3 实验环境

Table 3 Experiment environment

配置	参数
操作系统	Windows 7
开发语言及环境	Python 3.7.4/PyCharm 2019.2.1/Anaconda 3
CPU	Intel Xeon E5-2670 v3
内存/GB	4

4.2 参数设置

CTDNN-IPNS算法涉及参数较多,具体设置如下:

1)网络节点向量表示维度d:可根据实际需要选择任意维度,在本文实验中设置为128。

2)随机游走步长l:选择大于网络平均路径长度的数值,在本文实验中设置为10。

3)节点子序列截取考查范围w:在本文实验环境及数据集条件下设置为5。

4)信息传播时间范围Δt:根据网络信息传播特点设置该参数。通过对实验数据集的分析,在论文合作网络中,作者与其合作对象的合作时间一般约为3年,在电话通信网络中,一次信息传播的时间范围约为25 min,因此在本文实验中以3年和25 min设置该参数。

5)训练数据使用率γ:通常按照3∶1的比例将数据集划分为训练集和测试集,在本文实验中设置为0.75。

6)总游走次数:由于CTDNN-IPNS和CTDNE算法采用从随机选取的节点出发且依据指定规则进行随机游走的采样策略,而STWalk算法采取以网络快照内的每个节点为起点且依次开始随机游走的策略,为便于比较,在实验中将总游走次数设置为网络节点数的整数倍。

在实验中以总游走次数为变量开展算法性能测试,其中随机游走步长l、节点子序列截取考查范围w和传播时间范围Δt的敏感性见下文分析,而网络节点向量表示维度d和训练数据使用率γ的取值则采用经验值。

4.3 CTDNN-IPNS算法性能测试

为横向验证CTDNN-IPNS算法的性能,基于相同测试数据集,本文将CTDNN-IPNS算法与STWalk[16]和CTDNE[15]算法进行比较,对比算法采用清华大学发布的OpenNE框架内的相关函数进行实现。在测试过程中,网络节点向量表示维度$d=128$,节点子序列截取考察范围$w=5$,随机游走步长$l=10$,随机游走次数$n=30\,000$,训练数据使用率$\gamma=0.75$。

以DBLP数据集为例,CTDNN-IPNS、STWalk和CTDNE算法的动态网络表示学习结果经t-SNE算法[22]降维后的二维可视化效果如图3所示。可以看出,与STWalk和CTDNE算法相比,基于本文提出的节点采样策略,CTDNN-IPNS算法生成的动态网络表示学习结果能够更好地保持原有网络节点之间的内在相似性,数据集的6个类别在二维空间中的分布更集中,数量较少的黑色类别数据的聚集效果也更明显且各个类别的界限清晰,能够更好地支持后续节点的分类任务。

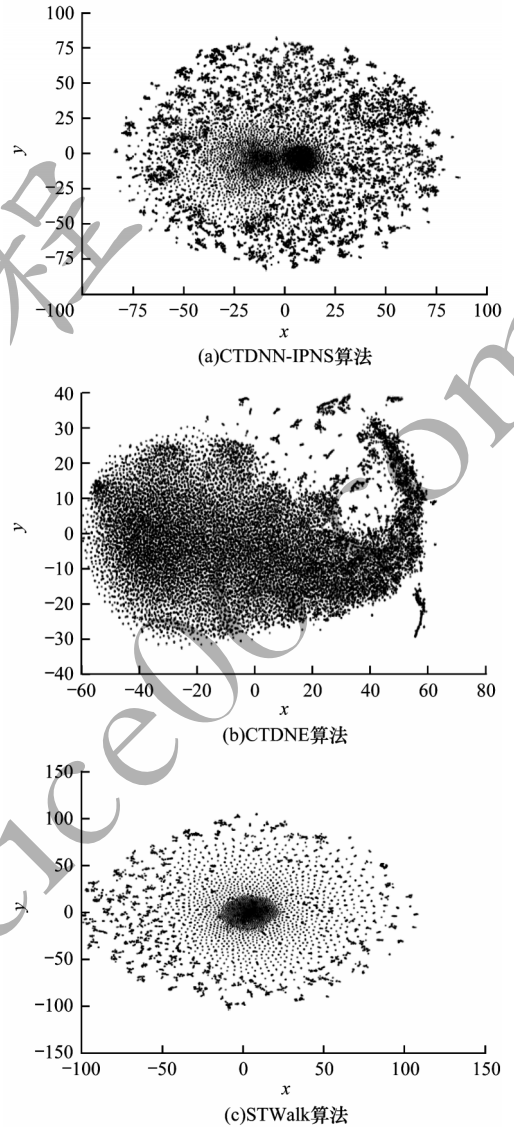


图3 3种算法的二维可视化效果

Fig.3 2D visualized effect of three algorithms

在总游走次数下,CTDNN-IPNS、CTDNE和STWalk算法对不同数据集的分类结果评价指标值(F1_micro和F1_macro)如表4~表6所示。上述分类结果评价指标值对应的曲线如图4~图6所示。根据上述分类结果的评价指标值可知,针对DBLP、AMiner和Reality-Call数据集,CTDNN-IPNS算法整体上优于STWalk和CTDNE算法。具体而言,在3组实验中,CTDNE算法分类结果的F1_micro和F1_macro值随节点采样次数的增加而呈现出上升趋势,但上升速度较慢。在对DBLP数据集和Aminer数据集进行节点分类时,CTDNN-IPNS算法分类结果的F1_micro和F1_macro值均为最高值,

且在总游走次数较少时,其优势更为明显。在对Reality-Call数据集进行分类时,3种算法均在总游走次数达到750以上时获得较好的分类效果,但CTDNN-IPNS算法的分类效果更佳,且在总游走次数低于750时,CTDNN-IPNS算法具有更好的分类性能,其F1_micro和F1_macro值更高且增速更快。

表4　CTDNN-IPNS、STWalk和CTDNE算法对DBLP数据集的分类结果评价指标值

Table 4　The evaluation index values of classification results on the DBLP dataset by CTDNN-IPNS, STWalk, CTDNE algorithm

总游走次数	CTDNE		STWalk		CTDNN-IPNS	
	F1_micro	F1_macro	F1_micro	F1_macro	F1_micro	F1_macro
31 572	0.255	0.096	0.444	0.406	0.611	0.593
63 144	0.331	0.212	0.444	0.416	0.617	0.604
94 716	0.408	0.326	0.446	0.408	0.605	0.588
126 288	0.472	0.413	0.422	0.384	0.609	0.592
157 860	0.498	0.453	0.444	0.398	0.599	0.586
189 432	0.525	0.494	0.419	0.383	0.605	0.588
221 004	0.552	0.525	0.402	0.357	0.604	0.590
252 576	0.556	0.536	0.411	0.362	0.592	0.581
284 148	0.588	0.569	0.434	0.386	0.584	0.568
315 720	0.595	0.580	0.416	0.371	0.596	0.586

表5　CTDNN-IPNS、STWalk和CTDNE算法对AMiner数据集的分类结果评价指标值

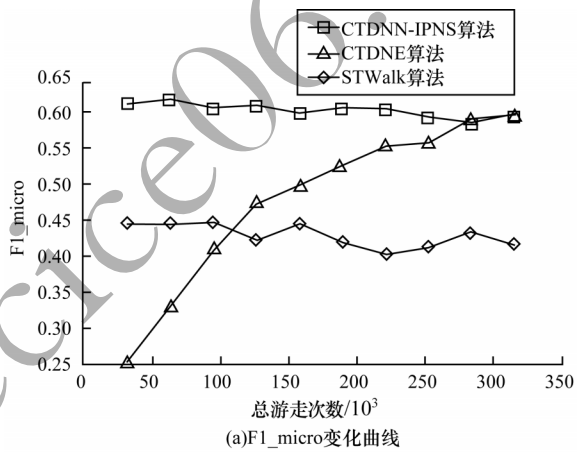
Table 5　The evaluation index values of classification results on the AMiner dataset by CTDNN-IPNS, STWalk, CTDNE algorithm

总游走次数	CTDNE		STWalk		CTDNN-IPNS	
	F1_micro	F1_macro	F1_micro	F1_macro	F1_micro	F1_macro
10 988	0.379	0.110	0.567	0.145	0.696	0.555
21 976	0.397	0.123	0.555	0.143	0.682	0.544
32 964	0.429	0.173	0.550	0.143	0.672	0.528
43 952	0.428	0.181	0.574	0.147	0.680	0.537
54 940	0.462	0.213	0.549	0.145	0.664	0.540
65 928	0.523	0.253	0.549	0.156	0.663	0.541
76 916	0.536	0.283	0.563	0.190	0.667	0.536
87 904	0.553	0.320	0.539	0.192	0.656	0.539
98 892	0.583	0.345	0.523	0.210	0.646	0.523
109 880	0.599	0.371	0.509	0.231	0.630	0.509

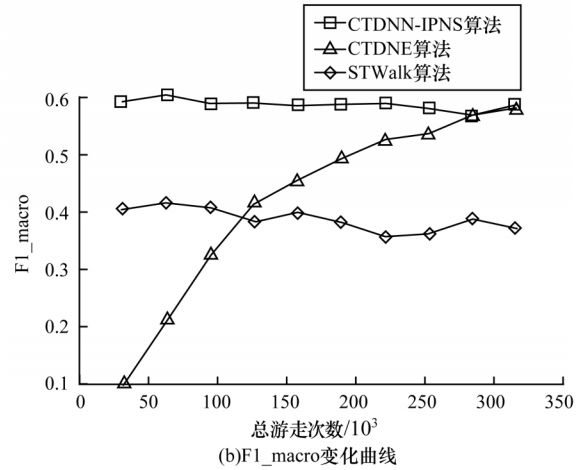
表6　CTDNN-IPNS、STWalk和CTDNE算法对Reality-Call数据集的分类结果评价指标值

Table 6　The evaluation index values of classification results on the Reality-Call dataset by CTDNN-IPNS, STWalk, CTDNE algorithm

总游走次数	CTDNE		STWalk		CTDNN-IPNS	
	F1_micro	F1_macro	F1_micro	F1_macro	F1_micro	F1_macro
200	0.600	0.375	0.897	0.892	0.439	0.305
400	0.633	0.605	0.897	0.896	0.957	0.954
600	0.900	0.893	0.931	0.930	1.000	1.000
800	0.980	0.979	0.897	0.896	0.980	0.976
1 000	0.961	0.961	0.931	0.927	0.960	0.954
1 200	1.000	1.000	0.897	0.895	1.000	1.000
1 400	0.981	0.960	0.897	0.895	0.960	0.960
1 600	1.000	1.000	0.862	0.861	0.980	0.980
1 800	1.000	1.000	0.931	0.930	1.000	1.000
2 000	1.000	1.000	0.828	0.827	1.000	1.000



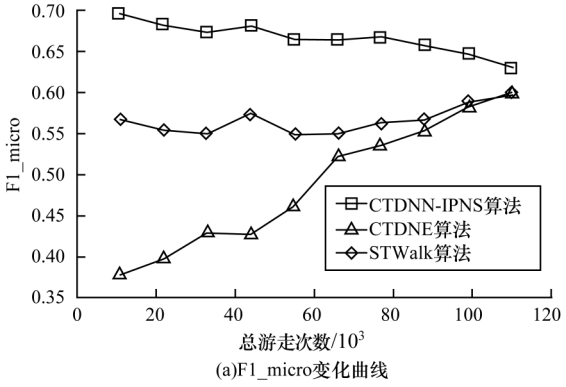
(a)F1_micro变化曲线



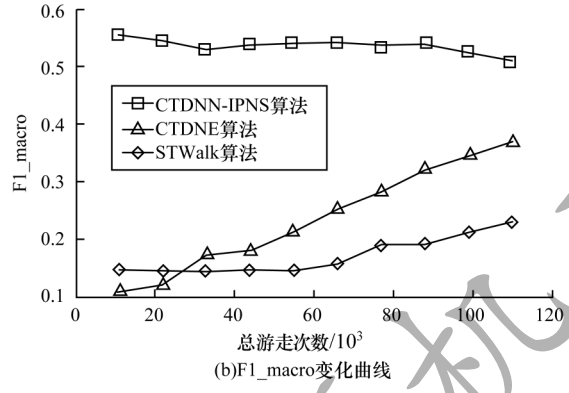
(b)F1_macro变化曲线

图4　3种算法对DBLP数据集的分类结果评价曲线

Fig.4　The evaluation curves of classification results on the DBLP dataset by three algorithms



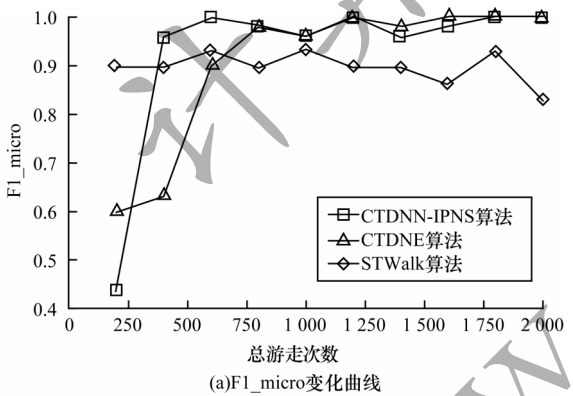
(a)F1_micro变化曲线



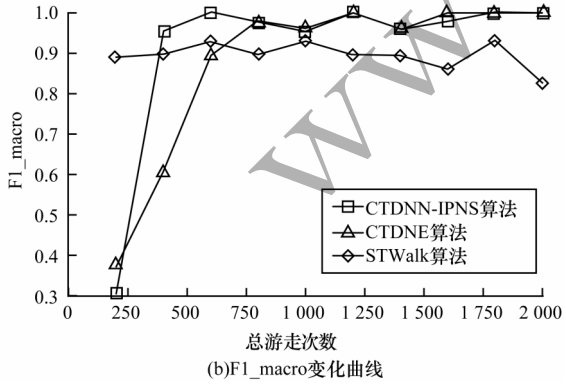
(b)F1_macro变化曲线

图5 3种算法对AMiner数据集的分类结果评价曲线

Fig.5 The evaluation curves of classification results on the AMiner dataset by three algorithms



(a)F1_micro变化曲线



(b)F1_macro变化曲线

图6 3种算法对Reality-Call数据集的分类结果评价曲线

Fig.6 The evaluation curves of classification results on the Reality-Call dataset by three algorithms

4.4 参数敏感性分析

随机游走步长l和节点子序列截取考查范围w及信息传播时间范围Δt是CTDNN-IPNS算法中的重要参数，本节通过在DBLP数据集上设定其他参数，分别改变l、w和Δt的取值大小来观察节点分类指标值（F1_micro和F1_macro）的变化情况，对算法的参数敏感性进行分析。如图7所示，随着l值的增加，F1_micro和F1_macro值先快速上升，再逐渐趋于平缓，曲线拐点在$l=7$附近，接近于DBLP数据集的平均路径长度值，且当l取值大于网络平均路径长度时，算法性能趋于平稳。这表明基于信息节点集的随机游走节点采样方式，能够较好地反映出网络的通联规律及其内在的结构属性。在本文实验中，为便于算法性能比较，将随机游走步长设定为3个数据集的网络平均路径最大值，故取$l=10$。在图8中，随着w值的增加，F1_micro和F1_macro值逐步提高，当$w\geqslant 5$时逐渐趋于平稳，因此在本文实验环境及数据集条件下取$w=5$。

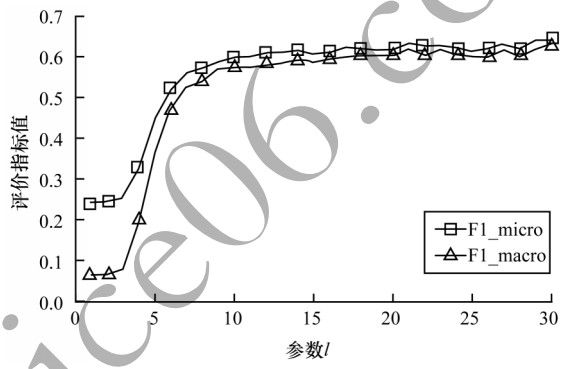


图7 CTDNN-IPNS算法分类性能随参数l的变化曲线

Fig.7 The change curves of classification performance of CTDNN-IPNS algorithm with parameter l

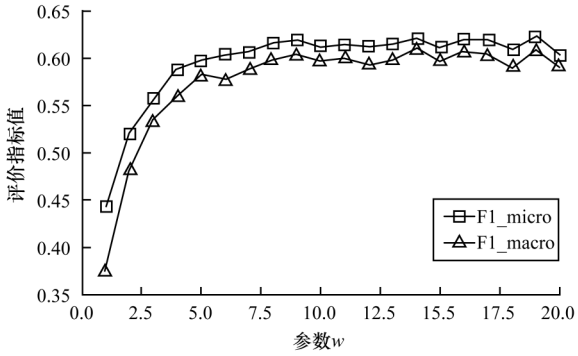


图8 CTDNN-IPNS算法分类性能随参数w的变化曲线

Fig.8 The change curves of classification performance of CTDNN-IPNS algorithm with parameter w

如图9、图10所示，在DBLP数据集和Reality-Call数据集的节点分类实验中，当Δt分别取3年和25 min时，算法分类性能较其他取值有小幅增长，这表明该参数的合理设置将直接影响算法的分类效果。

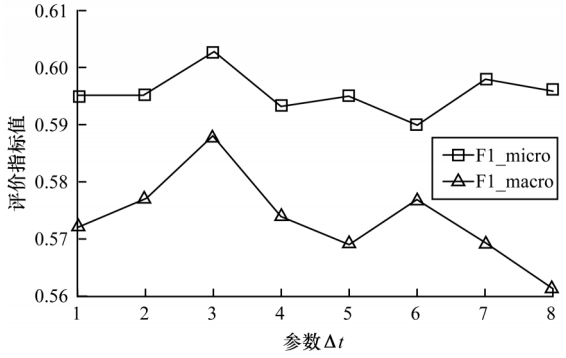


图9 CTDNN-IPNS算法在DBLP数据集上的分类性能随参数Δ*t*的变化曲线

Fig.9 The change curves of classification performance of CTDNN-IPNS algorithm on the DBLP dataset with parameter Δ*t*

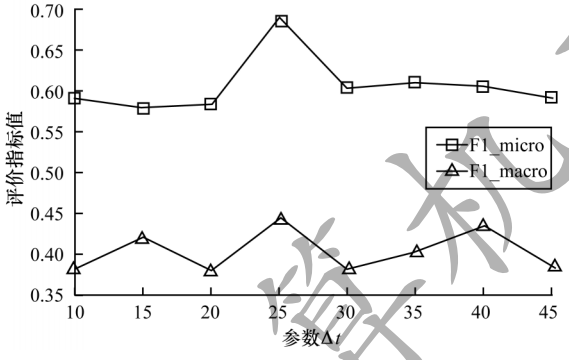


图10 CTDNN-IPNS算法在Reality-Call数据集上的分类性能随参数Δ*t*的变化曲线

Fig.10 The change curves of classification performance of CTDNN-IPNS algorithm on the Reality-Call dataset with parameter Δ*t*

4.5 结果分析

实验结果表明,在随机游走次数较少时,CTDNE算法因采用严格依照时间先后顺序的游走策略,在网络学习表示过程中受噪声影响较大,不能较好地捕捉到节点与同类别其他节点之间的关系,随着游走次数的增加,同类别节点的共现次数逐渐增加,其分类精度也随之提高。由于STWalk算法和CTDNN-IPNS算法在随机游走过程中,分别以节点历史邻居和信息传播集内节点为采样对象,因此在随机游走次数较少时表现出较好的分类性能,随着游走次数的增加,采集节点数逐渐增多,采样序列反而可能受到不同类别节点的干扰,导致分类性能略有下降,但总体表现基本平稳。

随着总游走次数的增加,CTDNE和CTDNN-IPNS算法的性能曲线逐渐趋同,这表明在总游走次数足够大的情况下,不同的随机游走策略最终反映出的图信息基本趋于一致,且对网络的整体表示学习能力相近,而STWalk算法侧重于关注单个网络快照上的节点,因此相比其他算法,整体分类性能相对较差。

5 结束语

本文提出一种新的连续时间动态网络节点分类算法,定义信息传播节点集,改进网络表示学习方法的节点序列采样策略,利用其生成的节点低维向量和LogicRegression分类器实现对连续时间动态网络的节点分类。实验结果表明,针对论文合作网络的作者分类和电话通信网络的用户分类问题,相比CTDNE和STWalk算法,该算法的网络表示学习结果能够更好地保留节点在原始网络结构上的内在相似性,且最终分类结果也更优。后续将结合节点属性、连边权重等信息,研究针对连续时间动态网络的分类算法,进一步提升其分类效果和适用范围。

参考文献

[1] TU Cunchao, YANG Cheng, LIU Zhiyuan, et al. Network representation learning: an overview[J]. Scientia Sinica Informationis, 2017, 47(8): 980-996. (in Chinese)
涂存超,杨成,刘知远,等. 网络表示学习综述[J]. 中国科学(信息科学),2017,47(8):980-996.

[2] BHAGAT S, CORMODE G, MUTHUKRISHNAN S. Node classification in social networks[J]. Computer Science, 2011, 16(3): 115-148.

[3] GOYAL P, FERRARA E. Graph embedding techniques, applications, and performance: a survey[EB/OL]. [2020-05-14]. https://arxiv. org/abs/1705. 02801.

[4] CUI Peng, WANG Xiao, PEI Jian, et al. A survey on network embedding[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(5): 833-852.

[5] OU Mingdong, CUI Peng, PEI Jian, et al. Asymmetric transitivity preserving graph embedding[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2016: 1105-1114.

[6] WANG Xiao, CUI Peng, WANG Jing, et al. Community preserving network embedding[C]//Proceedings of the 31st AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2017: 203-209.

[7] PEROZZI B, AL-RFOU R, SKIENA S. DeepWalk: online learning of social representations[C]//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2014: 701-710.

[8] GROVER A, LESKOVEC J. Node2Vec: scalable feature learning for networks[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2016: 855-864.

[9] WANG Daixin, CUI Peng, ZHU Wenwu. Structural deep network embedding[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2016: 1225-1234.

[10] CAO Shaosheng, LU Wei, XU Qiongkai. Deep neural networks for learning graph representations [C]// Proceedings of the 30th AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2016: 1145-1152.

[11] PEROZZI B, KULKARNI V, SKIENA S. Walklets: multiscale graph embeddings for interpretable network classification[EB/OL]. [2020-05-14]. https://arxiv. org/abs/1605. 02115.

[12] DONG Y, CHAWLA N V, SWAMI A. Metapath2Vec: scalable representation learning for heterogeneous networks[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York,USA:ACM Press,2017:135-144.

[13] KHOSHRAFTAR S, MAHDAVI S, AN A, et al. Dynamic graph embedding via LSTM history tracking [C]// Proceedings of 2019 IEEE International Conference on Data Science and Advanced Analytics. Washington D. C., USA: IEEE Press, 2019: 119-127.

[14] GOYAL P, KAMRA N, HE X R, et al. DynGEM: deep embedding method for dynamic graphs[EB/OL]. [2020-05-14]. https://arxiv. org/abs/1805. 11273.

[15] NGUYEN G H, LEE J B, ROSSI R A, et al. Continuous-time dynamic network embeddings[C]//Proceedings of International Conferences on World Wide Web. Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2018: 969-976.

[16] PANDHRE S, MITTAL H, GUPTA M, et al. STWalk: learning trajectory representations in temporal graphs[C]// Proceedings of ACM India Joint International Conference on Data Science and Management of Data. New York, USA: ACM Press, 2018: 210-219.

[17] CHENG X, JI L, YIN Y, et al. Network representation learning method based on spatial-temporal graph in dynamic network[C]//Proceedings of IEEE International Conference on Electronics Information and Emergency Communication. Washington D. C., USA: IEEE Press, 2019: 196-200.

[18] LEE J B, NGUYEN G, ROSSI R A, et al. Temporal network representation learning[EB/OL]. [2020-05-14]. https://arxiv. org/abs/1904. 06449.

[19] NGUYEN G H, LEE J B, ROSSI R A, et al. Dynamic network embeddings: from random walks to temporal random walks [C]//Proceedings of IEEE International Conference on Big Data. Washington D. C., USA: IEEE Press, 2018: 1085-1092.

[20] DALEY D J, KENDALL D G. Epidemics and rumours[J]. Nature, 1964, 204(4963): 1118.

[21] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C]//Proceedings of the 26th International Conference on Neural Information Processing Systems. New York, USA: ACM Press, 2013: 3111-3119.

[22] MAATEN L J, HINTON G E. Visualizing high-dimensional data using t-SNE[J]. Journal of Machine Learning Research, 2008, 9(2): 2579-2605.

编辑 陆燕菲