控制与决策
Control and Decision

一种基于节点嵌入表示学习的社区搜索算法

赵卫绩， 张凤斌， 刘井莲

引用本文:
赵卫绩, 张凤斌, 刘井莲. 一种基于节点嵌入表示学习的社区搜索算法[J]. 控制与决策, 2021, 36(8): 1970−1976.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2019.1439

一种基于节点嵌入表示学习的社区搜索算法

赵卫绩[1,2], 张凤斌[1†], 刘井莲[2,3]

(1. 哈尔滨理工大学 计算机科学与技术学院，哈尔滨 150080；2. 绥化学院 信息工程学院，
黑龙江 绥化 152061；3. 东北大学 计算机科学与工程学院，沈阳 110169)

摘 要: 针对已有社区搜索算法采用高维稀疏向量表示节点时间复杂度高的问题,提出一种基于节点嵌入表示学习的社区搜索算法CSNERL. 节点嵌入技术能够直接从网络结构中学习节点的低维实值向量表示,为社区搜索提供了新思路. 首先,针对已有节点嵌入算法存在较高概率在最亲近邻居间来回游走的问题,提出基于最亲近邻居但不立即回访随机游走的节点嵌入模型NECRWNR,采用NECRWNR模型学习节点的特征向量表示;然后,采用社区内所有节点的向量均值作为社区的向量表示,通过选择与当前社区距离最近的节点加入社区的方法实现一种新的社区搜索算法. 在真实网络和模拟网络数据集上分别与相关的社区搜索算法进行实验对比,结果表明所提出社区搜索算法CSNERL具有更高的准确性.

关键词: 社区搜索；节点嵌入；网络表示学习；社区发现；局部社区发现；随机游走

中图分类号: TP273 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2019.1439 **开放科学(资源服务)标识码(OSID):**

引用格式: 赵卫绩,张凤斌,刘井莲. 一种基于节点嵌入表示学习的社区搜索算法[J]. 控制与决策, 2021, 36(8): 1970-1976.

Community search algorithm based on node embedding representation learning

ZHAO Wei-ji[1,2], *ZHANG Feng-bin*[1†], *LIU Jing-lian*[2,3]

(1. School of Computer Science and Technology，Harbin University of Science and Technology，Harbin 150080，China；
2. School of Information Engineering，Suihua University，Suihua 152061，China；3. School of Computer Science and Engineering，Northeastern University，Shenyang 110169，China)

Abstract: Considering that the existing community search algorithms represent nodes as high-dimensional sparse vectors and have high time complexity, a community search algorithm based on node embedding representation learning (CSNERL) is proposed. Node embedding techniques can learn low-dimensional vectorial representation of nodes from network structure directly, and provide a new solution to community search problems. Firstly, in view of the problem that the existing node embedding algorithm has a high probability to walk back and forth between the closest neighbors, a node embedding model based on closest-neighbor biased random walk with non-immediately revisiting (NECRWNR) is proposed. Based on this model, vectorial representation of nodes is learned and used as feature vectors of nodes in the downsteam data mining task. Then, vectorial representation of a community is defined as the average of the vectors for nodes in the community, and a new community search algorithm is designed by choosing those nodes which are nearest to the current community. The proposed algorithm is tested on both real-world and synthetic network datasets with the related community search algorithms. The experimental results show that the CSNERL algorithm is more effective at community search than baselines.

Keywords: community search；node embedding；network representation learning；community detection；local community detection；random walk

0 引 言

社区结构是复杂网络的一个重要属性. 文献[1]将社区结构定义为网络中若干个内部连接紧密的节点的群组,节点与所在群组内的节点连接紧密,与其他群组连接稀疏[2]. 挖掘网络中的社区在个性化推荐、信息传播等方面都具有重要意义. 自2002年提

收稿日期: 2019-10-13；**修回日期:** 2020-02-18.

基金项目: 国家自然科学基金项目(61172168, 61772122, 61872074)；黑龙江省省属高校基本科研业务费科研项目(YWK10236200141).

†通讯作者. E-mail: zhangfb@hrbust.edu.cn.

出社区发现问题至今,涌现出了基于划分、模块度优化、标签传播等方法的众多社区发现算法,极大地推动了社区发现的发展[3]. 这些算法旨在寻找给定网络中的所有社区,但由于时间复杂度高而难以应用于大网络. 同时,许多应用场景也只是对网络中的某一个社区感兴趣,例如已知某研究领域的一名学者,寻找活跃在该领域的其他学者[4];对用户作个性化推荐时,了解其所在社区其他用户的喜好可以提供很多有益信息[5]. 不同于传统的社区发现方法,社区搜索[6-7]是从给定一个或多个节点出发,寻找包含它们的社区. 相比之下,社区搜索更关注给定节点周围的局部网络结构,能够高效返回用户所关心的社区.

近年来提出了很多社区搜索算法. Clauset[8] 定义了局部模块度R,通过选取使得R增量最大的外壳节点加入社区,实现了一种含节点集规模k约束的社区搜索算法. Huang等[9]提出基于节点相似度的社区质量度量函数tightness,在保证能够增加社区的tightness值的前提下,从外壳节点集中优先选择与当前社区内节点相似度最高的节点加入社区,计算给定节点所在的社区. 不同于文献[9]采用单层邻居节点集计算节点的相似度,Ma等[10]提出采用d-层邻居节点集计算节点相似度,Liu等[11]采用节点嵌入方法将节点映射到低维向量空间,在低维向量空间计算节点间的相似度. 此外,Panagiotakis 等[12]提出了基于流传播的社区搜索算法.

向量化网络数据是网络数据挖掘首要解决的问题. 传统方法是采用邻接矩阵存储网络,用邻接矩阵的行向量表示节点,但这种高维、稀疏的表示方法导致后续任务的时间复杂度较高[13]. 网络表示学习旨在从网络结构中学习节点的低维向量表示,在网络数据和已有数据挖掘模型之间搭起一座桥梁. 受自然语言处理领域词嵌入技术的启发, Perozzi等[14]将词嵌入技术引入网络表示学习,提出了第1个节点嵌入模型DeepWalk,引发了节点嵌入的研究热潮,此后相继涌现出LINE[15]、SDNE[16]、node2vec[17]、NEMCNB[11]等模型.

在此基础上,本文提出一种基于节点嵌入表示学习的社区搜索算法CSNERL (community search based on node embedding representation Learning). 首先,针对节点嵌入已有工作的不足,提出基于最亲近邻居但不立即回访随机游走的网络嵌入模型NECRWNR (node embedding model based on closest-neighbor biased random walk with non-immediately revisiting),基于该模型,为网络中每个节点学习一个低维实值向量; 然后,采用社区内所有节点的向量均值作为社区的向量表示,通过优先选择与社区距离最近的节点的方法逐渐扩展当前社区,提出基于社区向量表示的社区搜索算法CSCVR (community search based on community vector representation).

本文主要贡献如下: 1) 提出新的网络嵌入模型NECRWNR. 该模型既考虑了节点间亲近性对随机游走的影响,又避免了在最亲近邻居之间来回游走的问题,可以更好地保留原始网络的性质. 2) 提出的基于节点嵌入表示学习的社区搜索算法CSNERL是一个两阶段算法,集网络嵌入模型NECRWNR与社区搜索算法CSCVR于一体,用向量表示社区,将社区外节点与社区的距离建模为两个向量的内积运算,时间复杂度较低. 3) 在真实网络和模拟网络数据集上分别与相关的社区搜索算法和节点嵌入算法进行实验对比,结果表明所提出社区搜索算法CSNERL相比Clauset、GMAC、FlowPro、NEMCNB等基准算法能够得到更为准确的社区,此外,通过实验验证了NECRWNR模型有助于提高社区搜索结果的准确性.

1　基于节点嵌入表示学习的社区搜索算法

1.1　问题定义

用$G=(V,E)$表示网络. 其中:V为网络G中节点的集合;E为G中边的集合,$\Gamma(v)$为节点v的邻居节点的集合. 社区是网络中的一个密集子图,子图内节点连接紧密,子图间节点连接稀疏. 假设对网络G中的某一个社区C感兴趣,已知节点$v\in C$,社区搜索问题便是从网络G中寻找k个节点,使得这k个节点中有尽可能多的节点属于社区C.

基于节点嵌入表示学习的社区搜索问题可以分为两个子问题: 节点嵌入表示学习和基于节点向量表示的社区搜索.

定义1 (节点嵌入表示学习)[14-15]　给定网络$G=(V,E)$,节点嵌入的目标是学习一个映射函数$f: V\rightarrow R^d, d\ll|V|$,将网络$G$中的任一节点$u$映射到$d$维向量空间中的一个点$f(u)$,该$d$维空间仍保留节点在$G$中的性质.

定义2 (基于节点向量表示的社区搜索)　给定网络$G=(V,E)$以及节点在d维向量空间中的嵌入映射函数f,对于目标社区$C\subset G$,已知节点$v\in C$,给定参数k,社区搜索的目标是从节点v出发寻找k个节点,使得这k个节点中有尽可能多的节点属于目标社区C.

1.2　CSNERL算法描述

基于节点嵌入表示学习的社区搜索算法CSNERL是一个两阶段算法,集NECRWNR网络表

示与CSCVR社区搜索于一体,算法流程如图1所示.

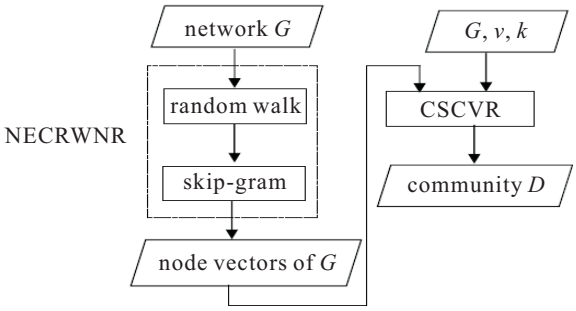

图 1 CSNERL算法流程

1.2.1 节点嵌入模型NECRWNR

NECRWNR与DeepWalk、node2vec、NEMCNB同属基于随机游走一类的节点嵌入模型. 首先给出随机游走过程的形式化描述,然后结合实例分析已有算法的不足.

假设从节点v_1出发长度为l的随机游走形成的节点路径表示为$[v_1, v_2, \ldots, v_{i-1}, v_i, v_{i+1}, \ldots, v_l]$,其中$v_i$为该路径对应的节点序列中的第$i$个节点. 不失一般性,将随机游走过程中刚访问过的节点v_i标记为u,则节点u的邻居节点集$\Gamma(u)$中的任一节点x成为下一个节点v_{i+1}的概率为

$$P(v_{i+1} = x | v_i = u) = \frac{w_{ux}}{z}. \tag{1}$$

其中: w_{ux}为边(u, x)的权重, z为归一化常数,计算为

$$z = \sum_{j \in \Gamma(u)} w_{uj}. \tag{2}$$

该类算法之间的主要区别在于计算w_{ux}的方法不同. DeepWalk算法将w_{ux}的值设置为1; node2vec算法认为w_{ux}与节点x和v_{i-1}之间的最短路径长度有关; NEMCNB算法采用Jaccard Index度量节点间的相似程度,认为w_{ux}与节点x和u之间的相似度有关,在社交网络中人们更倾向于选择与其关系最亲密的邻居节点.

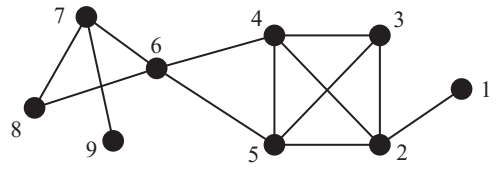

图 2 示例网络G

相比DeepWalk和node2vec, NEMCNB引入节点间相似性的差异指导随机游走,在社区搜索任务中取得了更好的实验结果,但在随机游走过程中存在着在最亲近邻居之间来回游走的问题. 结合图2说明该问题,边的权重w和游走到邻居节点的概率p的计算方法参见文献[11],这里只给出计算结果. 节点4的最亲近邻居是节点5,从节点4游走到节点5的概率最大, $p(5|4) = 0.406$. 节点5的最亲近邻居是节点4,从节点5游走到节点4的概率最大, $p(4|5) = 0.406$. 因此,在随机游走过程中从节点4出发,经过节点5又回到节点4的概率为0.165. 而在基于纯随机的游走方式中,由于节点4和节点5的度均为4,从节点4出发到节点5的概率为0.25; 同理,从节点5出发到节点4的概率也是0.25. 因此,从节点4出发经节点5再回到节点4的概率仅为0.062 5.

通过以上分析可以看出, NEMCNB算法偏向选择最亲近的邻居节点,出现在最亲近邻居间来回游走情况的概率较高. 针对此问题,提出基于最亲近邻居但不立即回访的随机游走CRWNR (closest-neighbor biased random walk with non-immediately revisiting),即当新选择的节点v_{i+1}与v_{i-1}相同时,放弃该次选择. 根据各邻居节点的概率分布,重新从当前节点v_i的邻居节点集中随机选择一个节点. 算法描述如下所示.

算法1 CRWNR.

输入: 网络$G = (V, E)$,起始节点u,行走的节点个数l;

输出: 节点路径p.

step 1: 初始化节点路径$p = [u]$.

step 2: 初始化变量$i = 1$.

step 3: 由式(1)计算从当前节点$p[i]$到其每一个邻居节点的概率.

step 4: 根据$\Gamma(p[i])$中每个节点的概率分布,随机选择一个节点,标记为y.

step 5: 如果节点y与$p[i-1]$是同一个节点,则重复执行step 4; 否则,将y添加到p中.

step 6: $i++$.

step 7: 重复执行$l-1$次step 3 ~ step 6.

step 8: 返回节点路径p,算法结束.

step 3 ~ step 5是本文算法区别于其他随机游走算法的主要之处. step 3和step 4实现了在随机游走过程中偏向最亲近邻居, step 5实现了随机游走过程中不立即回访.

在CRWNR的基础上,结合Skip-gram模型[14],提出节点嵌入模型NECRWNR. Skip-gram模型是自然语言处理中word2vec的一种实现方法,给定一条节点路径$[v_1, v_2, \ldots, v_{i-1}, v_i, v_{i+1}, \ldots, v_l]$,设定上下文的窗口大小为$w$,则节点$v_i$的上下文节点序列$C(v_i)$由节点路径中$v_i$前后各$w$个节点组成,即$C(v_i) = [v_{i-w}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+w}]$. 将节点看作单词,将节点路径看作句子,通过最大化目标函数

$$\max_f \sum_{v_i \in V} \log p(C(v_i)|f(v_i)) \tag{3}$$

学习节点嵌入映射函数f. 该目标函数是采用当前节点v_i的向量表示$f(v_i)$预测其上下文节点序列$C(v_i)$中节点的条件概率. NECRWNR算法描述如下所示.

算法2　NECRWNR.

输入: 网络$G=(V,E)$,从每一个节点出发的次数r,每次行走的节点个数l,上下文窗口大小w,节点向量的维度dn;

输出: 网络G的节点嵌入映射函数f.

step 1: 初始化节点路径序列ps = [ ].

step 2: 调用CRWNR(G,u,l)完成从V中的某一个节点u出发的随机游走,返回的节点路径标记为p.

step 3: 将p加入ps;

step 4: 每次从不同的节点出发,重复执行$|V|$次step 2和step 3.

step 5: 重复执行r次step 2 ~ step 4.

step 6: 利用ps中的节点路径构建语料库T.

step 7: 将T、w、dn作为输入,采用Skip-gram学习节点嵌入映射函数f,返回f,算法结束.

step 2 ~ step 5实现了从网络G中每一个节点重复进行r次基于最亲近邻居但不立即回访的随机游走,得到$|V|\times r$条节点路径,保存于ps中. 然后,将节点看作单词,节点路径看作句子,把ps转换为语料库T,利用Skip-gram算法学习节点嵌入映射函数f.

1.2.2　基于社区向量表示的社区搜索算法CSCVR

社区是网络中内部连接紧密的节点的群组. 节点嵌入方法学习的是节点的向量表示,为了度量节点与社区之间的距离,首先给出社区的向量表示方法.

定义3 (社区的向量表示)　在节点向量表示的基础上,借鉴多维空间中几何中心的思想,提出采用社区内节点的向量均值作为社区的向量表示. 对于社区D,D内节点个数用$|D|$表示,则社区D的向量表示$f(D)$定义为

$$f(D)=\frac{1}{|D|}\sum_{u\in D} f(u). \tag{4}$$

在社区向量表示的基础上,给出节点与社区的距离定义.

定义4 (节点与社区的距离)　在$f(D)$的基础上,节点v与社区D的距离dis(D,v)定义为

$$\mathrm{dis}(D,v)=1-f(D)\cdot f(v), \tag{5}$$

其中运算符$\cdot$为向量的内积运算,其取值范围为[0, 1],描述的是社区D与节点v之间的相似程度. 因此,dis(D,v)反映节点v与社区D之间的距离远近. 通过优先选择与社区连接且距离最近的节点加入目标社区的方法得到新的社区搜索算法CSCVR. 算法描述如下所示.

算法3　CSCVR.

输入: 网络$G=(V,E)$,节点嵌入映射函数f,初始节点v,节点个数k;

输出: 最有可能是节点v所在社区的k个节点的集合D.

step 1: 初始化$D=\{v\}$以及其外壳节点集$N=\Gamma(v)$.

step 2: 初始化社区D的向量表示$f(D)=f(v)$.

step 3: 由式(5)计算N中每一个节点到社区D的距离,选择距离D最近的节点,标记为y.

step 4: 将节点y加入D中.

step 5: 如果D内节点个数为k,则转step 9.

step 6: 更新D的向量表示.

step 7: 更新外壳节点集N.

step 8: 重复执行step 3 ~ step 7,直至N为空.

step 9: 返回社区D,算法结束.

step 3 ~ step 8通过从社区D的外壳节点集N中选择距离D最近的节点加入社区D的方法扩展目标社区D,直至D内的节点个数达到k或者外壳节点集N为空.

节点y加入目标社区D后,在step 6更新社区D的向量表示. 可用式(7)快速计算加入节点y后的新社区D'的社区向量表示$f(D')$,有

$$f(D')=\frac{|D|f(D)+f(y)}{|D|+1}. \tag{6}$$

step 7的更新外壳节点N,可通过增量计算完成N的更新. 由于节点y加入到社区D中导致外壳节点集N发生变化,更新后的外壳节点集为$N'=(N\setminus\{y\})\bigcup(\Gamma(y)\setminus D)$.

算法3的计算主要集中在step 3 ~ step 8的$k-1$次循环,每次循环涉及到$|N|$个外壳节点与社区距离的计算. 由于社区和节点都采用dn维向量表示,计算外壳节点与社区距离的时间复杂度为$O(\mathrm{dn})$. 假设节点的平均度为d_{avg},那么第i次循环外壳节点个数的上限为$i\times d_{\mathrm{avg}}$,对应的时间复杂度为$O(i\times d_{\mathrm{avg}}\times \mathrm{dn})$. $k-1$次循环的时间复杂度总计为$O(k^2\times d_{\mathrm{avg}}\times \mathrm{dn})$.

2　实验与结果分析

2.1　实验设定

2.1.1　数 据 集

分别采用真实网络和模拟网络测试算法的有效性. 真实网络包括: Zachary空手道俱乐部成员关系网

络Karate[18]、海豚网络Dolphins[19]、美国大学生橄榄球网络Football[1]、美国政治书籍网络Polbooks[20]. 模拟网络由Lancichinetti等[21]提出的LFR模型生成,该模型的主要参数包括网络中的节点个数n、节点平均度$d_{\rm avg}$、节点最大度$d_{\max}$、混合参数mu. mu值越大,意味着生成网络中的节点有越多的邻居节点在其所在社区之外,正确识别节点所在社区的难度也就越大. 本文生成3组网络数据集,每组包含10个网络. LFR模型的参数设置情况如表1所示.

表1 LFR模型的参数值

group name	n	$d_{\rm avg}$	$d_{\max}$	mu
L10K	10000	10	50	0.05, 0.10, . . ., 0.50
L30K	30000	10	50	0.05, 0.10, . . ., 0.50
L50K	50000	10	50	0.05, 0.10, . . ., 0.50

2.1.2 对比算法及度量指标

为验证CSNERL算法的有效性,与Clauset[8]、GMAC[10]、FlowPro[12]、NEMCNB[11]进行比较. 此外,与DeepWalk[14]、node2vec[15]和NEMCNB[11]比较以验证NECRWNR模型的有效性.

采用准确率(P)、召回率(R)和F指标(F_1)[9-11]衡量算法的优劣. 评估算法在数据集上的表现时,对数据集中的每个节点进行一次实验,取n(n为节点个数)次实验的平均值作为最后结果. 根据k值设定不同,分两种情况度量算法优劣: 1) 同一个起始节点相同k值时,采用召回率R值衡量各种算法的有效性,召回率R值越大,表明算法越好. 2) 同一个起始节点不同k值时,k值的大小将导致P值和R值产生不同的变化趋势,为了平衡P值和R值,采用F_1指标来度量算法的有效性,F_1指标取值越大,表明算法越好.

2.2 社区搜索算法比较

在该实验中,设置k值为起始节点所在真实社区中节点的个数. 首先在真实网络数据集上进行对比实验,实验结果如表2所示.

表2 各算法在真实网络数据集上的对比结果

R	karate	dolphins	football	polbooks
Clauset	0.8817	0.8841	0.7464	0.7646
GMAC	0.5611	0.6803	0.4808	0.4890
FlowPro	0.7908	0.8889	0.7678	0.7566
NEMCNB	0.5903	0.3754	0.8893	0.4759
CSNERL	**0.9739**	**0.9923**	**0.9163**	**0.7949**

由表2可见,CSNERL算法在4个数据集上均取得了最高的召回率R值,表现最好,其次是Clauset和FlowPro算法.

在LFR模拟网络数据集上进行对比实验. 由于FlowPro算法时间复杂度高,L30K、L50K两组数据集不与该算法进行对比,比较结果如表3所示.

表3 各算法在LFR网络数据集上的对比结果

R	Clauset			GMAC			FlowPro			NEMCNB			CSNERL		
	L10K	L30K	L50K	L10K	L30K	L50K	L10K	L30K	L50K	L10K	L30K	L50K	L10K	L30K	L50K
mu = 0.05	0.9140	0.9127	0.9117	0.8806	0.8801	0.8779	0.9695	–	–	0.9708	0.9556	0.9515	**0.9991**	**0.9992**	**0.9993**
mu = 0.10	0.8183	0.8128	0.8119	0.8846	0.8847	0.8866	0.9628	–	–	0.9733	0.9513	0.9480	**0.9983**	**0.9985**	**0.9990**
mu = 0.15	0.7249	0.7180	0.7197	0.8755	0.8751	0.8764	0.9599	–	–	0.9616	0.9444	0.9414	**0.9975**	**0.9983**	**0.9987**
mu = 0.20	0.6399	0.6333	0.6316	0.8680	0.8596	0.8614	0.9521	–	–	0.9527	0.9361	0.9318	**0.9958**	**0.9980**	**0.9977**
mu = 0.25	0.5647	0.5534	0.5526	0.8694	0.8476	0.8411	0.9396	–	–	0.9402	0.9215	0.9163	**0.9932**	**0.9948**	**0.9941**
mu = 0.30	0.4863	0.4866	0.4875	0.8612	0.8273	0.8225	0.9212	–	–	0.9225	0.9029	0.9008	**0.9849**	**0.9835**	**0.9847**
mu = 0.35	0.4206	0.4253	0.4227	0.8276	0.8156	0.8026	0.8879	–	–	0.8901	0.8816	0.8752	**0.9649**	**0.9709**	**0.9720**
mu = 0.40	0.3632	0.3618	0.3598	0.8251	0.7902	0.7827	0.8521	–	–	0.8516	0.8500	0.8433	**0.9369**	**0.9446**	**0.9430**
mu = 0.45	0.3117	0.3054	0.3049	0.7898	0.7683	0.7574	0.7974	–	–	0.8089	0.8064	0.8062	**0.8997**	**0.9015**	**0.9056**
mu = 0.50	0.2471	0.2493	0.2480	0.7305	0.7350	0.7209	0.7198	–	–	0.7467	0.7430	0.7477	**0.8347**	**0.8337**	**0.8442**

由表3可以得到如下结论: 1) 随着mu值增大,各种算法呈现下降的趋势,这也验证了LFR网络中mu值越大社区发现的难度越高这一先验性质. 2) 算法在mu值相同但网络规模不同的数据集上的召回率R值差别很小,表明这些算法具有良好的稳定性. 3) 在3组LFR网络数据集上,CSNERL算法均取得了最高的R值,实验结果最好,其次是NEMCNB、FlowPro、GMAC算法. 相比之下,基于局部模块度优化的Clauset算法在$\rm mu \geqslant 0.10$的数据集上表现最差,且mu值越大与其他算法之间的差别也越大.

2.3 节点嵌入方法比较

分别用DeepWalk、node2vec和NEMCNB学习节点向量,然后采用基于社区向量表示的社区搜索算法CSCVR进行社区搜索,以此对比各节点嵌入模型在

社区搜索问题中的有效性. LFR网络数据集上的实验结果如表4所示.

由表4可以看出: 1) 节点嵌入方法结合基于社区向量表示的社区搜索算法CSCVR在LFR网络上取得了很好的实验结果. 这一方面有CSCVR算法的贡献, 同时也与节点嵌入算法能够保留原始网络的性质有关系. 2) 对比各节点嵌入算法的表现可以看出, NECRWNR表现最好, 其次是NEMCNB、node2vec, 相比之下DeepWalk表现最差. 这与DeepWalk采用纯随机的游走方式, 没有考虑节点间的相似度有关. NECRWNR优于NEMCNB主要源于考虑了节点间的亲近性对随机游走的影响, 避免了在最亲近邻居之间来回游走的策略.

表 4　节点嵌入方法在LFR网络数据集上的对比结果

R	DeepWalk			node2vec			NEMCNB			NECRWNR		
	L10K	L30K	L50K	L10K	L30K	L50K	L10K	L30K	L50K	L10K	L30K	L50K
mu = 0.05	0.9844	0.9958	0.9959	0.9864	0.9940	0.9938	0.9896	0.9960	0.9961	**0.9991**	**0.9992**	**0.9993**
mu = 0.10	0.9783	0.9919	0.9931	0.9814	0.9928	0.9933	0.9885	0.9945	0.9959	**0.9983**	**0.9985**	**0.9990**
mu = 0.15	0.9712	0.9883	0.9886	0.9801	0.9884	0.9917	0.9866	0.9925	0.9951	**0.9975**	**0.9983**	**0.9987**
mu = 0.20	0.9577	0.9852	0.9861	0.9733	0.9864	0.9882	0.9855	0.9877	0.9889	**0.9958**	**0.9980**	**0.9977**
mu = 0.25	0.9376	0.9811	0.9818	0.9710	0.9849	0.9858	0.9843	0.9858	0.9865	**0.9932**	**0.9948**	**0.9941**
mu = 0.30	0.9078	0.9491	0.9539	0.9540	0.9751	0.9770	0.9758	**0.9838**	**0.9849**	**0.9849**	0.9835	0.9847
mu = 0.35	0.8767	0.9160	0.9244	0.9306	0.9562	0.9577	0.9610	0.9625	0.9637	**0.9649**	**0.9709**	**0.9720**
mu = 0.40	0.8418	0.8846	0.8922	0.9027	0.9316	0.9353	0.9318	0.9357	0.9387	**0.9369**	**0.9446**	**0.9430**
mu = 0.45	0.7985	0.8334	0.8447	0.8589	0.8891	0.8961	0.8935	0.8964	0.8983	**0.8997**	**0.9015**	**0.9056**
mu = 0.50	0.7305	0.7797	0.7877	0.8011	0.8277	0.8372	0.8233	0.8299	0.8376	**0.8347**	**0.8337**	**0.8442**

2.4　运行时间比较

在L10K数据集上进行算法运行时间对比实验. 对数据集中每个节点进行一次实验, 合计进行10万次实验, 取平均运行时间进行比较. 所有实验均在Windows10操作系统下, 由Python编码完成, 实验所使用的硬件参数是: Intel (R) Core (TM) i7-7 700 CPU @3.60 GHz, 内存DDR4 16 G. 比较结果如图3所示.

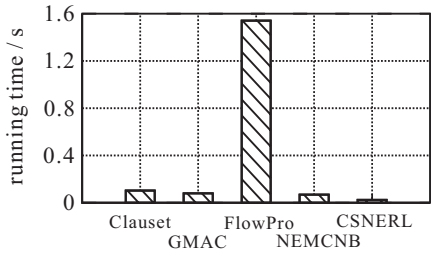

图 3　社区搜索算法在L10K上的平均运行时间

由图3可见, 除FlowPro进行一次社区搜索平均用时1.54 s外, 其他算法用时最多的是Clauset的0.10 s. 而CSNERL算法平均用时0.02 s, 远远低于其他4种社区搜索算法.

2.5　参数讨论

讨论参数k的变化对算法CSNERL的影响. 在L10K(mu = 0.5)网络上分别使k取节点所在真实社区节点个数n的0.6、0.8、1.0、1.2、1.4倍进行实验. 由于要对算法在不同k值的表现进行比较, 用F_1值衡量算法的好坏. 实验结果如图4所示.

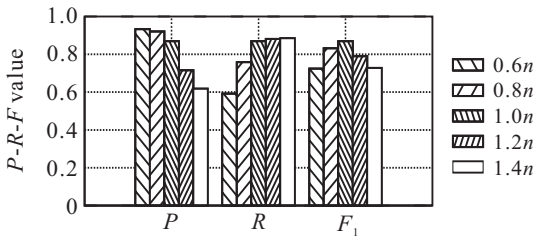

图 4　L10K (mu = 0.50)数据集上不同k值的实验结果

由图4可见, 在k逐渐增大的过程中, 准确率P值逐渐减小, 而召回率R值逐渐增大, F指标F_1值呈现出先增大后减少的趋势. 较大k值的返回结果是在较小k值返回结果的基础上又多返回了一些节点, 因此查全率逐渐增大. 由于算法返回的节点个数越来越多, 返回结果中位置靠后的节点距离起始节点越来越远, 准确率P值逐渐减小, 而F_1值在k等于真实社区节点个数时获得最大值.

3　结　论

随着互联网和物联网的快速发展, 产生了更大规模的网络数据, 传统的全局社区发现方法无法有效处理这些大网络. 社区搜索算法由于所需的时间复杂度与所在网络的规模大小没有直接关系, 引起了学者们的广泛关注. 受深度学习在网络表示学习方向的启发, 本文提出了一种基于节点嵌入表示的

两阶段社区搜索算法CSNERL,并通过实验完成了对CSNERL算法以及NECRWNR模型对社区搜索有效性的验证. 此外,观察了参数k的变化对CSNERL算法的影响,为参数设置提供了依据.

参考文献(References)

[1] Girvan M, Newman M E J. Community structure in social and biological networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2002, 99(12): 7821-7826.

[2] 邓琨, 李文平, 陈丽, 等. 一种新的基于标签传播的复杂网络重叠社区识别算法[J]. 控制与决策, 2020, 35(11): 2723-2732.
(Deng K, Li W P, Chen L, et al. A novel algorithm for overlapping community detection based on label propagation in complex networks[J]. Control and Decision, 2020, 35(11): 2723-2732.)

[3] Fortunato S, Hric D. Community detection in networks: A user guide[J]. Physics Reports, 2016, 659: 1-44.

[4] Kloumann I M, Kleinberg J M. Community membership identification from small seed sets[C]. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2014: 1366-1375.

[5] Zhao Y L, Nie L Q, Wang X Y, et al. Personalized recommendations of locally interesting venues to tourists via cross-region community matching[J]. ACM Transactions on Intelligent Systems & Technology, 2014, 5(3): 1-26.

[6] Sozio M, Gionis A. The community-search problem and how to plan a successful cocktail party[C]. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington: ACM, 2010: 939-948.

[7] Cui W Y, Xiao Y H, Wang H X, et al. Local search of communities in large graphs[C]. Proceedings of the ACM SIGMOD International Conference on Management of Data. Snowbird: ACM, 2014: 991-1002.

[8] Clauset A. Finding local community structure in networks[J]. Physical Review E, 2005, 72(2): 026132.

[9] Huang J B, Sun H L, Liu Y G, et al. Towards online multiresolution community detection in large-scale networks[J]. Plos One, 2011, 6(8): e23829.

[10] Ma L H, Huang H, He Q M, et al. GMAC: A seed-insensitive approach to local community detection[C]. Proceedings of 15th International Conference on Data Warehousing and Knowledge Discovery. Prague: Springer, 2013: 297-308.

[11] Liu J D, Wang D L, Feng S, et al. Learning distributed representations for community search using node embedding[J]. Frontiers of Computer Science, 2019,13(2): 437-439.

[12] Panagiotakis C, Papadakis H, Fragopoulou P. Local community detection via flow propagation[C]. Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. Paris: ACM, 2015: 81-88.

[13] Cui P, Wang X, Pei J, et al. A survey on network embedding[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(5): 833-852.

[14] 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: ACM, 2014: 701-710.

[15] Tang J, Qu M, Wang M, et al. LINE: Large-scale information network embedding[C]. Proceedings of the 24th International Conference on World Wide Web. Florence: ACM, 2015: 1067-1077.

[16] Wang D X, Cui P, Zhu W W. Structural deep network embedding[C]. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco: ACM, 2016: 1225-1234.

[17] Grover A, Leskovec J. Node2vec: Scalable feature learning for networks[C]. Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco: ACM, 2016: 855-864.

[18] Zachary W W. An information flow model for conflict and fission in small groups[J]. Journal of Anthropological Research, 1977, 33(4): 452-473.

[19] Lusseau D, Schneider K, Boisseau O J, et al. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations — Can geographic isolation explain this unique trait?[J]. Behavioral Ecology and Sociobiology, 2003, 54(4): 396-405.

[20] Newman M E J. Modularity and community structure in networks[J]. Proceedings of the National Academy of Sciences, 2006, 103(23): 8577-8582.

[21] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms[J]. Physical Review E, 2008, 78(4): 046110.

作者简介

赵卫绩(1980−), 男, 副教授, 博士生, 从事社区发现、数据挖掘等研究, E-mail: sdzhaoweiji@163.com;

张凤斌(1965−), 男, 教授, 博士生导师, 从事网络安全、入侵检测技术等研究, E-mail: zhangfb@hrbust.edu.cn;

刘井莲(1980−), 女, 副教授, 博士生, 从事社会网络分析、社区发现等研究, E-mail: datamining@163.com.

(责任编辑：郑晓蕾)