

基于多查询的社交网络关键节点挖掘算法

辛国栋, 朱滕威, 黄俊恒, 魏家扬, 刘润萱, 王巍

(哈尔滨工业大学(威海)计算机科学与技术学院, 山东 威海 264209)

摘要: 关键节点挖掘是复杂网络领域的研究重点和热点。针对社交网络中关键嫌疑人挖掘问题, 提出基于多查询的社交网络关键节点挖掘算法。该算法将已知嫌疑人作为查询节点, 提取其所在的局部拓扑结构, 并计算局部拓扑结构中非查询节点的关键程度, 从中选择关键程度较高的节点进行推荐。针对现有方法中关键节点计算复杂度高、已知查询节点信息难以有效利用的问题, 提出一个两阶段的基于多查询的社交网络关键节点挖掘算法, 整合多查询节点的局部拓扑信息和全局节点聚合特征信息, 将计算范围从全局缩减到局部, 进而对相关节点的关键程度进行量化。具体而言, 利用带重启策略的随机游走算法获得多个查询节点的局部拓扑结构; 为了得到节点的嵌入向量, 基于 graphsage 模型构建一种无监督的图神经网络模型, 该模型结合节点的自身特征和邻居聚合特征来生成嵌入向量, 从而为算法框架的相似度计算提供信息输入。基于与查询节点特征的相似性, 衡量局部拓扑中节点的关键程度。实验结果显示, 所提算法在时间效率和结果有效性方面均优于传统关键节点挖掘算法。

关键词: 社交网络; 随机游走; 图神经网络; 节点嵌入向量; 关键节点

中图分类号: TN92

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2024013

Multi-query based key node mining algorithm for social networks

XIN Guodong, ZHU Tengwei, HUANG Junheng, Wei Jiayang, Liu Runxuan, WANG Wei

School of Computer Science and Technology, Harbin Institute of Technology(Weihai), Weihai 264209, China

Abstract: Mining key nodes in complex networks has been a hotly debated topic as it played an important role in solving real-world problems. However, the existing key node mining algorithms focused on finding key nodes from a global perspective. This approach became problematic for large-scale social networks due to the unacceptable

收稿日期: 2023-02-01; 修回日期: 2023-11-21

通信作者: 王巍, wwht@hit.edu.cn

基金项目: 国家自然科学基金(62272129); 国家重点研发计划(2021YFB2012400); 中央高校基本科研业务费专项(HIT.NSRIF.2020098)

Foundation Items: The National Natural Science Foundation of China (62272129), The National Key R&D Program of China(2021YFB2012400), The Fundamental Research Funds for the Central Universities (HIT.NSRIF.2020098)

引用格式: 辛国栋, 朱滕威, 黄俊恒, 等. 基于多查询的社交网络关键节点挖掘算法[J]. 网络与信息安全学报, 2024, 10(1): 79-90.

Citation Format: XIN G D, ZHU T W, HUANG J H, et al. Multi-query based key node mining algorithm for social networks[J]. Chinese Journal of Network and Information Security, 2024, 10(1): 79-90.

storage and computing resource overhead and the inability to utilize known query node information. A key node mining algorithm based on multiple query nodes was proposed to address the issue of key suspect mining. In this method, the known suspects were treated as query nodes, and the local topology was extracted. By calculating the critical degree of non-query nodes in the local topology, nodes with higher critical degrees were selected for recommendation. Aiming to overcome the high computational complexity of key node mining and the difficulty of effectively utilizing known query node information in existing methods, a two-stage key node mining algorithm based on multi-query was proposed to integrate the local topology information and the global node aggregation feature information of multiple query nodes. It reduced the calculation range from global to local and quantified the criticality of related nodes. Specifically, the local topology of multiple query nodes was obtained using the random walk algorithm with restart strategy. An unsupervised graph neural network model was constructed based on the graphsage model to obtain the embedding vector of nodes. The model combined the unique characteristics of nodes with the aggregation characteristics of neighbors to generate the embedding vector, providing input for similarity calculations in the algorithm framework. Finally, the criticality of nodes in the local topology was measured based on their similarity to the features of the query nodes. Experimental results demonstrated that the proposed algorithm outperformed traditional key node mining algorithms in terms of time efficiency and result effectiveness.

Keywords: social network, random walk, graph neural network, node embedding vector, key node

0 引言

在复杂网络分析中, 关键节点挖掘成为热点研究方向, 受到诸多研究者的关注。关键节点挖掘也被称为节点中心性分析^[1], 该任务的目标是从社交网络中挖掘有重要影响力的节点或子网络之间的桥梁节点。关键节点挖掘在很多领域具有重要的应用价值, 如挖掘新浪微博等社交媒体中的意见领袖节点, 有助于社交媒体的舆情监控; 从网络中识别并保护 hub 节点和枢纽节点, 有利于维持网络平稳安全地运行。

现有关键节点挖掘算法的主要研究思路是寻找全局视角下的关键节点, 对于大规模社交网络来说需要的存储和计算资源开销是难以接受的, 并且这些算法无法利用已知的查询节点信息。例如, 在犯罪网络中, 初始线索往往只掌握有限个嫌疑人, 传统的侦破过程只根据整个网络中具有重要影响力的人物或者与已知嫌疑人直接关联的人物进行推荐, 而这些推荐的人物也往往并非与已知嫌疑人相关的关键人物。关键节点查询示例如图 1 所示, 节点 1、节点 2、节点 3 表示 3 个已知嫌疑人, 传统算法认为节点 4、节点 7、节点 11 这 3 个节点关键程度更高, 但是中间虚线框中 4 个节点与所有已知节点有很强的联系, 是本文

所关注的关键节点, 传统算法因为不能利用查询节点的信息而挖掘不出真正和已知查询节点相关的关键节点。

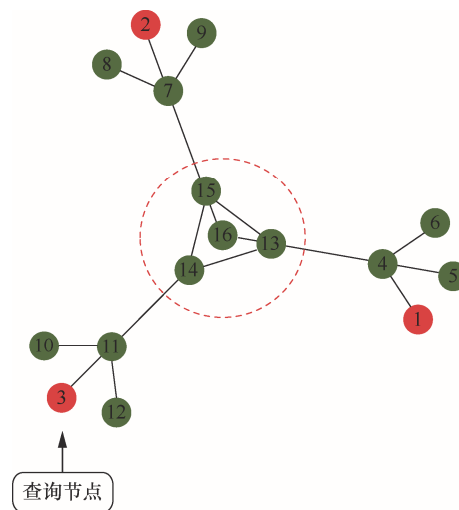


图 1 关键节点查询示例
Figure 1 Example of key nodes query

本文关注的基于已知查询节点挖掘关键节点的任务与传统的键节点挖掘任务有以下几点不同。① 对关键节点的定义不同。传统的关键节点是指相对网络其他节点而言能够在更大程度上影响网络结构与功能的一些特殊节点, 而本文关注的关键节点是与已知查询节点相关性较高的节

点。② 任务目标不同。传统的关键节点挖掘任务是找到全局结构上的关键节点, 而本文关注查询节点所在的局部结构上的关键节点。③ 任务设计思路不同。传统的关键节点挖掘任务大多是基于网络结构特性设计的, 如基于节点度中心性的算法、基于网络中最短路径的算法。本文关注基于查询节点所在局部结构设计的关键节点挖掘算法。

针对这个任务, 本文提出一种两阶段的基于多查询的社交网络关键节点挖掘算法——MQ-KNM 算法, 主要思想是整合多查询节点所在的局部拓扑结构信息和全局节点聚合特征信息来挖掘查询节点所在局部网络中的关键节点, 本文主要贡献如下。

1) 提出基于多查询节点的局部关键节点挖掘问题并对其进行建模。

2) 设计一个两阶段的 MQ-KNM 算法, 在提高算法时间效率的同时, 有效挖掘出查询节点所在局部结构中的关键节点。

3) 对第一阶段随机游走向量的更新机制进行了增强, 缩短了算法运行时间; 在第二阶段设计一种无监督的图神经网络模型。

4) 在真实数据上的实验结果表明, 本文提出的 MQ-KNM 算法相对传统关键节点算法执行速度更快, 结果的准确性更高, 证明了两阶段算法框架的有效性, 为后续的相关研究奠定了基础。

1 相关工作

关键节点挖掘算法分为以下四大类^[1]。

第一类是基于节点特性的算法, 该类算法关注节点邻居的数量和节点在网络中的位置, 是最简单直观的关键节点挖掘方法。度中心性^[2]是其中比较有代表性的算法, 度中心性用节点邻居数量的多少来表示该节点在网络中的重要程度。K-shell^[3]则通过递归地删除网络中度数等于 k 的节点, 并将节点 KS 值赋为 k , 节点的 KS 值越大, 在网络中的关键程度越高。

第二类是基于路径的算法, 该类方法使用节点间的路径信息来刻画节点的重要性。这一思想打破了使用节点度作为评价指标的局限性, 能够挖掘到在实际网络中度数较小但比较重要的节点。其中, 与节点最短路径相关的典型算法有离

心中心性^[4]和接近中心性^[5], 其分别通过节点间的最短路径长度和最短路径长度平均值来衡量节点的重要程度。而 Newman^[6]根据网络中随机游走生成的路径经过节点的次数来表示一个节点的重要性, 依据这一思路提出了基于随机游走的介数中心性算法。

第三类是基于特征向量的算法, 此类算法不仅考虑邻居节点的数量, 还考虑邻居节点的质量。这方面主要研究有 PageRank^[7]算法和 LeaderRank^[8]算法以及相关的变体。该类算法广泛应用于网页重要性排序任务, 将网页视作节点, 网页之间的链接关系可以构建一个网络, 最终为每个节点计算出一个分数来衡量该节点的关键程度。

第四类是基于节点移除与收缩的方法, 这类方法与系统科学中确定系统核心的思路类似, 认为节点的重要性往往体现在该节点被移除之后对网络的破坏性上。陈勇等^[9]在通信网络数据集上, 通过移除节点和相关边后得到图生成树的数目来表示节点的重要性; 谭跃进等^[10]将节点收缩后的网络聚集度作为节点重要性的评估标准。此类算法计算复杂度较高, 难以应用到实际场景。

近几年, 陆续出现了 OlapGN^[11]、FIP^[12]等社区重叠的方法, 来判断节点的影响力。Aghaalizadeh 等^[13]提出了三阶段社区检测算法, 根据本地相似性标准选择重要节点来扩展初始社区, 直到所有节点都成为其中一个社区的成员。最后, 将发现的社区合并, 形成最终的社区结构。

总体来说, 这些算法大多使用网络中节点、路径等网络特性来衡量节点的关键程度, 并且这些算法是从全局角度来计算的, 导致上述算法不能有效利用已知的查询节点信息。复杂网络的小世界^[14]特性表明节点所在的局部拓扑信息可以替代全局网络拓扑信息, 本文基于这种特性提出的 MQ-KNM 算法将关键节点的计算限制在查询节点所在的局部网络结构中, 通过节点嵌入向量余弦距离的计算可以衡量任意节点间的相似度。这两点改进使该算法框架能够有效利用已知查询节点信息, 克服现有关键节点挖掘方法不能利用查询节点信息的缺陷。

2 MQ-KNM 算法

本节针对 MQ-KNM 算法的两阶段流程分别设计相应的算法。第一阶段设计了局部拓扑搜索 (LTS, local topology search) 算法, 通过已知查询节点, 生成其所在的局部拓扑结构; 第二阶段设计了一种无监督的图神经网络模型来计算局部拓扑结构中节点的嵌入向量。之后使用两阶段算法得到的信息来量化局部拓扑中节点的关键程度。

2.1 问题定义

定义 1 无向网络。一个无向网络记为 $G=(V,E)$, 其中 $V=\{v_1, v_2, \dots, v_n\}$ 为网络中的节点集合, $E=\{e_1, e_2, \dots, e_m\}$ 为网络中边的集合, n 和 m 分别代表节点数和边数。

定义 2 局部拓扑结构。在一个无向图 $G=(V,E)$ 中, 给定一个查询节点集合 $Q \subseteq V$ 、评价函数 f , 对于 G 的任一子图结构 $H=(V_H, E_H)$, 其中 $V_H \subseteq V$ 和 $E_H \subseteq E$, 如果 H 满足以下 3 个要求:

- 1) Q 是 V_H 的子集;
- 2) H 是一个连通子图;
- 3) $f(H)$ 值是 G 的所有可行子图结构中最优的。

这里的子图 H 就定义为查询节点集合 Q 的局部拓扑结构。

假设多查询节点不在同一连通图上的情况不具有一般性, 不在讨论范围内, 本文只考虑查询节点集合 Q 在同一个连通图上的情况。

定义 3 基于查询的关键节点。给定图 $G=(V,E)$ 和查询节点集合 Q 及其所在的局部拓扑结构 $H=(V_H, E_H)$, 计算局部拓扑结构中任意节点 $v \in V_H$ 与 Q 中节点相似度的平均值, 按照相似度均值大小排序, 得到排序后的局部关键节点集合 $S(s_i > s_j, \forall i < j)$ 。

2.2 算法总体框架

本文提出的 MQ-KNM 算法包括独立的两个阶段: ① 局部拓扑搜索阶段; ② 基于图神经网络的节点嵌入向量学习阶段。最终利用两阶段得到的信息来挖掘关键节点。算法总体框架如图 2 所示。

图 2 的左侧为框架第一阶段, 设计局部拓扑搜索算法, 基于随机游走算法^[15]生成局部拓扑结

构; 右侧为框架第二阶段, 设计一个基于 graphsage 模型^[16]的无监督图神经网络 (GNN, graph neural network), 用来生成节点的嵌入表示。在上述两阶段的基础上, 融合查询节点所在的局部拓扑结构及其节点的嵌入向量, 并计算与多查询节点的特征相似性对局部拓扑结构中节点排序获得关键节点。

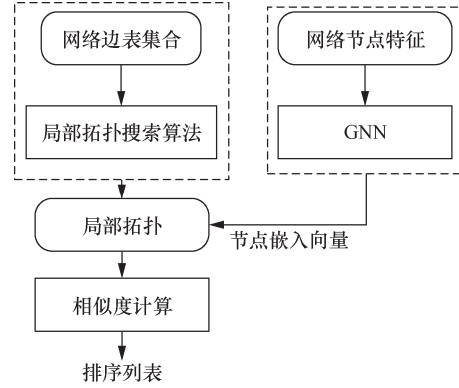


图 2 算法总体框架

Figure 2 The overall framework of the algorithm

2.3 局部拓扑搜索算法

局部拓扑搜索算法采用随机游走的方式生成查询节点所在的局部拓扑结构。随机游走被广泛用于解决复杂网络抽样问题, 随机游走算法的基本思想是: 选择网络上的某个节点作为起始点, 每次迭代随机选择一个相邻节点, 直到达到终止条件, 最终完成对网络节点的采样。本文将查询节点作为随机游走算法的起始节点, 使用最终采样出的节点来生成局部拓扑结构。

在基于多查询节点的任务中, 多个查询节点通常在结构上是邻近的, 以查询节点作为起始节点, 随着迭代次数的增长, 上述随机游走算法访问的节点会扩散到距离查询节点较远的节点上, 这些节点往往是无用节点, 一般通过减少迭代次数来解决, 但这会使得对查询节点周围节点采样不充分。与传统的随机游走算法不同, 本文使用带重启策略的多起始节点的随机游走算法, 将可能游走到的节点限制在查询节点周围, 使得最终生成的局部拓扑结构节点间的连通性更强。

给定查询节点集合 Q , 首先将随机游走过程的起始向量初始化为 $\mathbf{X}^{(0)} = \mathbf{e}_Q$, 其中, $\mathbf{e}_Q$ 为 one-hot 向量, $|\mathbf{e}_Q| = n$ 表示向量长度, n 为网络节

点个数。 e_Q 中对应查询节点的位置为1，其余为0。在得到起始向量后，使用图邻接矩阵信息提取节点概率转移矩阵；使用节点概率转移矩阵更新起始向量，并迭代这一过程，直到达到终止条件，再对终止向量中的访问概率按照大小排序。最后，根据所设置的评价函数选择部分访问概率高的节点构建局部拓扑结构。LTS算法步骤如图3所示，详细流程如下。

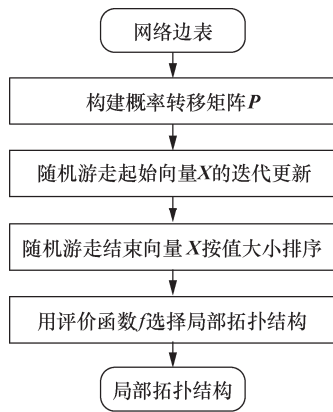


图3 LTS算法步骤
Figure 3 LTS algorithm step

(1) 构建概率转移矩阵

随机游走算法是从起始节点开始，通过概率转移矩阵来选择游走到的邻居节点的。概率转移矩阵 P 表示图 G 中节点间的转移概率，它的每一项都是表示概率的非负实数， P_{ij} 表示节点 i 到节点 j 的转移概率，该矩阵每列的和为1，表示当前节点转移到其他节点的概率和为1。对于无向图来说， $P_{ij} = P_{ji}$ ，下面是概率转移矩阵的生成过程。

节点数为 n 的图 $G=(V,E)$ 的邻接矩阵记为 $A(G)=[a_{ij}]_{n \times n}$ ，若 v_i 与 v_j 邻接， $a_{ij}=1$ ，否则为0。对邻接矩阵 A 执行列归一化操作生成节点的概率转移矩阵，每个元素为 $P_{ij} = \frac{W_{ij}a_{ij}}{\sum_n W_{in}a_{in}}$ ， $1 \leq i \leq n, 1 \leq j \leq n$ 。本文算法假设节点之间的转移概率是等权重的，即权重矩阵 W 为单位矩阵，若节点 i 和 j 之间有连边，则 P_{ij} 为节点 i 度的倒数，否则为0。

(2) 起始向量的迭代更新

在给定概率转移矩阵 P 后，每次迭代过程使用 P 更新起始向量，直到达到设置的迭代次数，

最终得到包含每个节点的访问概率的终止向量，迭代过程如式(1)所示。

$$X^{(T)} = \alpha P X^{(T-1)} + (1-\alpha) X^{(0)}, T \geq 1 \quad (1)$$

其中， T 为迭代次数， P 为转移概率矩阵， $1-\alpha$ 为重启概率， $X^{(T)}$ 表示从起始节点出发，经过 T 次迭代，游走到图上任意节点的期望概率。 $X^{(0)}$ 为 $1 \times n$ 维起始向量。

(3) 选择局部拓扑结构

给定查询节点集 Q 和终止向量，为了构建其局部拓扑结构 H ，本文采用文献[17]中的处理过程，并将评价函数 f 定义为局部拓扑结构 H 中所有节点的最小度。以往的研究结果表明，最小度作为局部拓扑的评价指标有多种优势：

1) 最小度是复杂网络的基本特征，可以用来描述局部拓扑连接的紧密程度；

2) 文献[18]将最小度与连通性和网络直径等指标进行了对比，发现最小度在社交网络分析中是一个比较好的度量；

3) 文献[19]将最小度与平均度和节点密度等指标进行比较，实验结果表明，在局部拓扑搜索的任务中，最小度指标性能较优。

评价函数 f 的计算如式(2)所示。

$$f(H) = \min_{v \in H} \deg H(v) \quad (2)$$

其中， $f(H)$ 为局部拓扑结构 H 对应的评价函数值，公式右半部分表示求 H 中任意节点 v 的最小度。

LTS算法首先对随机游走终止向量 $X^{(T)}$ 中各节点的访问概率进行排序；然后依次选择前 $k(1 \leq k \leq n)$ 个节点构成局部拓扑结构 H_k ，根据评价函数计算相应的 $f(H_k)$ 值，如果 $f(H_k)$ 相同，则选择节点最少的 H_k ；最终得到查询节点所在的局部拓扑结构 H_k 。

LTS算法的实现过程如算法1所示，算法的第3)~第4)行对应起始向量的迭代更新过程；第6)~第11)行对应局部拓扑结构的选择过程。

算法1 LTS算法

输入 邻接矩阵 A ，查询节点集合 Q ，重启概率 $1-\alpha$ ，循环次数 T ，评价函数 f

输出 局部拓扑结构 H_k

1) $P = \text{getTransMatrix}(A)$ //生成概率转移矩阵

```

2)  $X = \text{initWalker}(Q)$ //根据  $Q$  初始化起始向量
3) for  $t=0$  to  $T$ :
4)    $\text{update}(X, 1-\alpha)$ //根据式(1)
5) end for
6)    $X = \text{rank}(X)$ //对向量  $X$  元素大小排序
7)  $\text{minDegree}=0$ 
8) for  $k=0$  to  $n$ :
9)    $H_k = \text{topk}(X, k)$ //构建局部拓扑结构
10)   $\text{minDegree} = \max(\text{value}, f(H_k))$ 
11) if  $\text{minDegree} \geq 1$ ://最小度大于 1
12)   break
13) end for
14) 返回  $H_k$ 

```

2.4 基于图神经网络的节点嵌入向量学习算法

图神经网络模型通过聚合邻居特征来得到节点嵌入向量, 可以充分利用节点的特征信息。嵌入向量不仅可以表征节点及其邻居的特征, 还可以在两个节点没有直接相邻的情况下, 通过计算嵌入向量之间的距离来衡量任意节点之间的相似度, 相似度高的节点之间联系更密切。本文基于 **graphsage** 模型构建一种无监督的图神经网络模型, 该模型结合了节点的自身特征和邻居聚合特征来生成节点的嵌入向量, 从而为算法框架的最后一步提供信息输入。

基于 **graphsage** 模型的无监督图神经网络框架如图 4 所示, 包括 5 层结构: 输入层、聚合层、全连接层、内积层和输出层。训练完成后, 将全连接层的输出作为节点的嵌入向量。

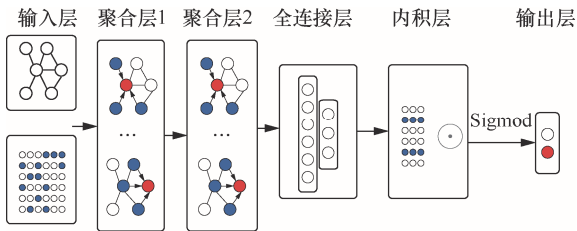


图 4 基于 **graphsage** 模型的无监督图神经网络框架
Figure 4 An unsupervised GNN framework based on **graphsage** model

2.4.1 各层神经网络设计

模型的前半部分包括输入层、聚合层和全连接层, 用于与节点邻居特征聚合。将社交网络节点的特征信息和邻接矩阵信息输入模型的聚合层, 在 $k(k=2)$ 个聚合层中生成节点聚合特征,

并经过全连接层对上一层输出的聚合特征做加权。

输入层为节点的特征向量, 如式(3)所示。

$$h_0 = [X, A] \quad (3)$$

其中, h_0 表示输入层的输入向量, X 为节点特征向量, A 为图邻接矩阵。

聚合层为模型的核心, 也叫聚合函数。节点聚合函数有很多种, 包括求均值、取最大值等, 不同函数的区别主要在于对邻居节点特征向量的聚合方式。为了平衡训练效率和结果准确度, 以及在节点特征的聚合中综合考虑邻居节点的信息, 而不过分关注个别节点。均值聚合能够取得较好的性能, 兼顾了模型训练效率和关键节点挖掘的准确性。本文采用均值聚合的方式, 并对节点自身特征向量和邻居特征向量执行连接操作, 得到节点的聚合特征。聚合层的计算如式(4)所示, 其中聚合函数 **AGGREGATE** 如式(5)所示。

$$h_1 = \text{AGGREGATE}^k(h_0) \quad (4)$$

$$e_v^k = \text{CONCAT}(\{W_1 \cdot e_v^{k-1}, \text{MEAN}\{W_2 \cdot e_u^{k-1}, \forall u \in N(v)\}\}) \quad (5)$$

其中, h_0 是模型的输入向量, v 为当前节点, $N(v)$ 为节点 v 的邻居节点, k 为聚合层的数量, 该模型 k 的取值为 2。 W_1 和 W_2 分别代表当前节点特征和邻居聚合特征的权重矩阵。

全连接层如式(6)所示。

$$h_2 = h_1 W_3 + b \quad (6)$$

其中, h_1 是经过聚合层后输出的节点聚合特征向量, W_3 为全连接层权重矩阵, b 为偏置系数。

模型的后半部分是内积层和输出层, 如式(7)所示。模型学习过程中随机选择两个节点计算其嵌入向量的内积, 通过激活函数输出最终的预测结果。

$$\hat{y} = \sigma(h_2(\text{src}) \cdot h_2(\text{dst})) \quad (7)$$

其中, h_2 是将节点聚合特征向量降低到固定维度, $\hat{y}$ 为最终预测标签, $\cdot$ 为内积符号, σ 为 sigmoid 激活函数。

2.4.2 损失函数

本文选用交叉熵损失函数作为该模型的目标函数, 如式(8)所示。

$$\text{loss} = -\hat{y} \log(y) - (1 - \hat{y}) \log(1 - y) \quad (8)$$

其中, loss 为损失函数值, $\hat{y}$ 为最终预测标签, y 为真实标签。

为了使相邻的节点嵌入向量的相似度更高, 在构建训练样本时, 如果两个节点在网络中邻接, 则真实标签记为 1, 否则为 0。训练过程中使用的样本形如(节点 1, 节点 2, 标签)。对于每一个训练样本, 计算节点 1 和节点 2 的嵌入向量的内积, 使用 sigmoid 函数构建输出层, 预测每个节点对所对应的标签。计算出损失函数后执行反向传播, 更新模型参数。

2.5 关键节点挖掘

基于多查询的关键节点挖掘任务使用前两个阶段的结果, 即局部拓扑结构和节点的嵌入向量来计算最终的关键节点。为了量化局部拓扑结构中节点的关键程度, 本文提出一种基于节点相似度的关键节点计算方法。

为了衡量任一节点与查询节点集合的相似度, 本文采用余弦相似度来衡量节点嵌入向量的相似度。相似度越大, 表示该节点与多个查询节点的相关性越高, 其关键程度就越高。具体计算如式(9)所示。

$$S_i = \sum_{j \in Q} \cos(\text{emb}_i, \text{emb}_j) \quad (9)$$

其中, S_i 代表图 G 任意节点 i 与查询节点集合的相似度, Q 为查询节点集合, emb_i 和 emb_j 分别代表节点 i 和节点 j 的嵌入向量, $\cos$ 表示计算两个嵌入向量的余弦相似度。

3 实验与分析

本节对所提算法进行验证, 首先给出实验数据, 然后分别对 MQ-KNM 算法的第一阶段算法和最终产生的关键节点结果设置对比实验。实验环境为 Windows10、Python3.6、PyCharm。

3.1 实验数据

本文选取数据集 cit-DBLP、soc-BlogCatalog^[20] 和 email-EU, 分别代表 3 种不同规模的社交网络。下面分别从节点数、边数、平均度和 triangle 数等统计指标的角度对数据集进行介绍, 其中节点数和边数是网络的基本信息, 而平均度和 triangle 数则分别衡量了网络的连接密度和结构的稳定

性。3 个数据集的统计信息如表 1 所示。

表 1 3 个数据集的统计信息
Table 1 Statistics of the three datasets

数据集	节点数	边数	平均度	triangle 数
cit-DBLP	12.6K	49.7K	7	134K
soc-BlogCatalog	88.8K	2.1M	47	153.6M
email-EU	32.4K	54.4K	3	147K

cit-DBLP 数据集是基于 DBLP 数据库构造的论文引用网络, 节点数为 12 600 个, 边数为 49 700, 节点平均度为 7, 网络中的 triangle 数为 134 000 个。

soc-BlogCatalog 数据集是 BlogCatalog 上博主与朋友之间关系网络, 网络节点有 88 800 个, 边数为 2 100 000 个, 平均度为 47, triangle 数为 153 600 000 个, 在节点数目为 email-EU 数据集 2.5 倍的情况下, triangle 数多了 1 044 倍, 表明 BlogCatalog 数据集的网络结构稳定, 节点连接密度较高。

email-EU 数据集是欧洲邮件网络, 原始数据是包括发送方、接收方和时间戳的三元组, 本文只保留发送方和接收方的连边关系, 不使用时间戳数据。网络节点数为 32 000 个, 边数为 54 000 个, 节点平均度为 3, 相比 cit-DBLP 数据集连接关系较弱, triangle 数为 147 000。

3.2 对比算法

本文算法的对比算法同样在第一阶段生成局部拓扑结构, 然后对其中的节点按照下面 5 种中心性指标分别计算出节点的关键程度, 相应设置 5 个对比算法, 并对其结果排序, 生成关键节点列表。

1) DC (degree centrality)^[21]。DC 是根据邻居数量度量节点重要性最简单的方法。假设图 G 的节点数为 n , 最终计算出的 DC 值通过除以 $n-1$ 来正则化。

2) SOC (second order centrality)^[22]。SOC 为给定节点的二阶中心性定义为在图 G 上随机游走到该节点的返回时间的标准差。二阶中心性越低, 表明该节点的中心性越高。

3) BC (betweenness centrality)^[23]。BC 为图 G 中任意节点 v 的中介中心性, 定义为通过节点

v 的任意两节点之间最短路径的个数。

4) VR (vote rank) [24]。根据投票方案计算图 G 中节点的排名,最初是为了找到网络中重要的传播者节点。所有节点都对相邻节点进行投票,迭代选择投票最高的节点,被选中的节点在下一轮的投票能力被降低。

5) EC (eigenvector centrality) [25]。EC 为特征向量中心性根据节点邻居的中心性计算当前节点的中心性。节点 i 的特征向量中心性是图 G 邻接矩阵的特征向量的第 i 个元素。

3.3 实验结果与分析

实验部分首先对不同的随机游走类型所得到的局部拓扑结构的连通性和算法运行效率进行对比实验,然后在第一阶段生成查询节点的局部拓扑结构的基础上,将本文算法与多种社交网络中心性算法所挖掘出的关键节点进行对比,并对结果在部分数据集上做可视化展示。

3.3.1 评价指标

衡量图连通程度最常用的评价指标为连通度,定义为图中节点的最小度。为了对第一阶段 LTS 算法生成的局部拓扑结构的连通性进行评价,综合衡量计算时间效率与效果,本文选用 ANC (average node connectivity) [26] 作为评价指标,ANC 定义为网络中所有节点对之间不相交路径数的平均值。最小度是衡量网络中最差的连通情况,而 ANC 使用平均值使得结果更加平滑,更能反映网络中的连接状况。

通过计算准确率对 MQ-KNM 算法和其他中心性算法产生的关键节点进行评价,准确率的计算方式为:计算生成的 top- K 节点中与查询节点关联的比例,如果该节点在任意查询节点对的路径上,则视作与查询节点关联。

3.3.2 不同随机游走算法实验分析

为了验证本文算法的有效性和时间效率,本节对 LTS 算法所采用的随机游走算法进行实验对比。

(1) 局部拓扑结构连通性分析

将 LTS 算法中采用的 MRWR (multi-agent random walk with restart) 算法与其余 3 种随机游走算法进行对比:MRW (multi-agent random walk)、SRW (single random walk)、SRWR (single

random walk with restart) [27]。SRW 算法为每个初始节点生成单独的起始向量,游走过程中依次更新,最终将所有向量累加在一起;而 MRW 算法将多个初始节点融合到单个起始向量中,对多个查询节点同时更新。SRWR 算法和 MRWR 算法分别是 SRW 算法和 MRW 算法对应的带重启策略的版本,本文第一阶段算法采用 MRWR 算法生成查询节点所在的局部拓扑结构。实验参数设置查询节点个数为 2,迭代次数 T 为 10。分别在 cit-DBLP、email-EU、soc-Blog Catalog 数据集上设置对比实验,实验结果如图 5~图 7 所示。

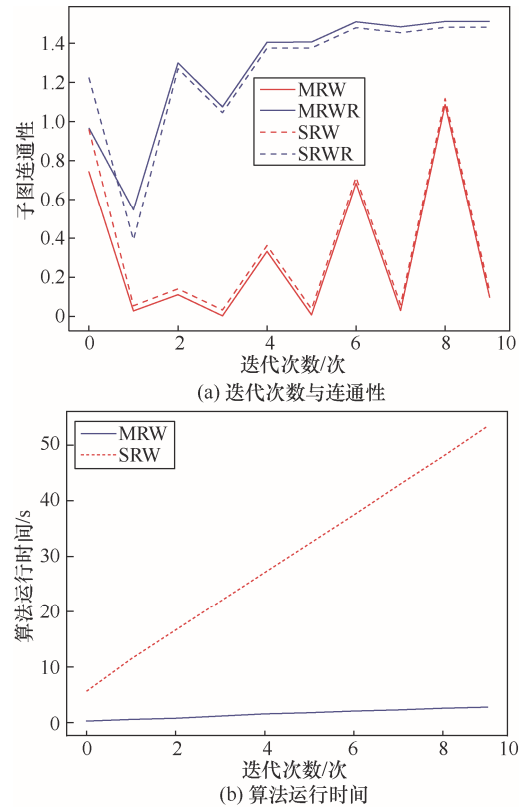


图 5 cit-DBLP 数据集实验结果
Figure 5 Experimental results on the cit-DBLP dataset

如图 5(a)、6(a)、7(a)所示,SRW 算法和 MRW 算法最终的子图连通性都收敛于 0,而 SRWR 算法和 MRWR 算法经过 10 次迭代以后保持稳定。说明带重启策略的 SRWR 算法和 MRWR 算法最终生成的局部拓扑结构中节点间的连接密度比 SRW 算法和 MRW 算法更高,生成的局部拓扑结构更加稳定。从不同数据集的实验结果来看:

soc-BlogCatalog 数据集子图连通性结果比较稳定,而其他两个数据集每次迭代的结果震荡较为严重。这主要是受数据集对应网络中节点连接关系的影响, soc-BlogCatalog 数据集的平均度较高,子图中节点之间 triangle 数较多,使得结果更加稳定,而另外两个数据集上节点连接关系较弱,迭代过程中容易震荡。从以上不同数据集的实验结果可以看出, MRWR 和 SRWR 两种随机游走算法的 ANC 值较高,生成的局部拓扑结构连通性较好。

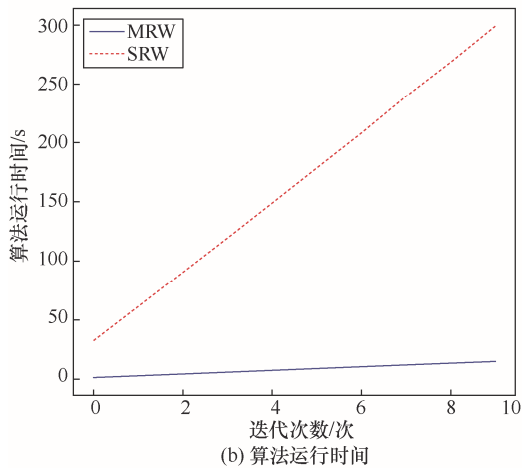
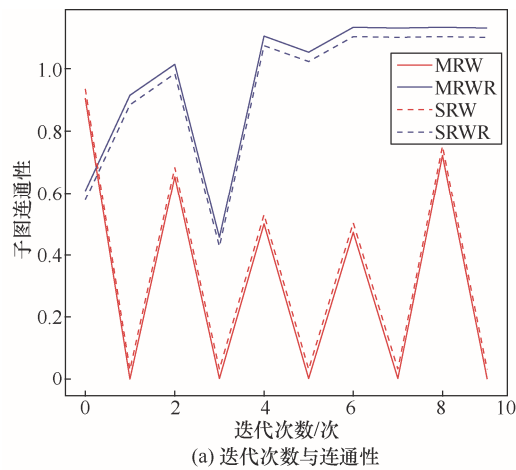


图6 email-EU 数据集实验结果

Figure 6 Experimental results on the email-EU dataset

(2) 算法运行效率分析

算法运行时间结果如图 5(b)、6(b)、7(b)所示, MRW 算法和 MRWR 算法比 SRW 算法和 SRWR 算法运行速度快。这是因为 SRW 算法和 SRWR 算法为每个查询节点生成单独的起始向量,随着查询节点个数的增加,要生成的起始向量的数目

随之增多,算法运行时间跟着增加,在大规模数据集上运行此类算法往往是不现实的。由此看出,在产生的局部拓扑连通性效果相近的情况下,本文所使用的 MRWR 算法的运行时间远远小于 SRWR 算法。

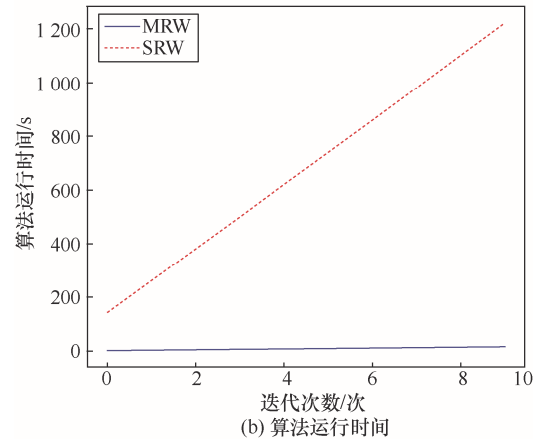
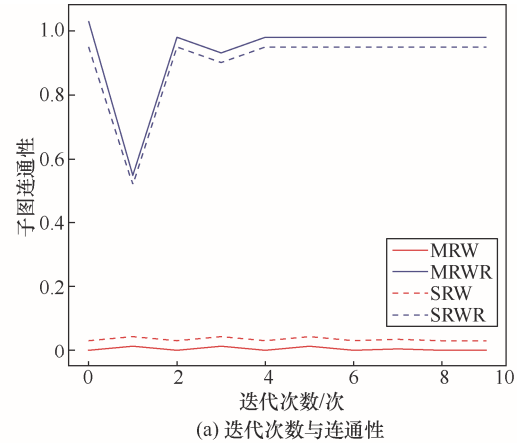


图7 soc-BlogCatalog 数据集实验结果

Figure 7 Experimental results on the soc-BlogCatalog dataset

(3) 查询节点个数对实验结果的影响

为了分析查询节点个数对 LTS 算法生成的局部拓扑结构连通性的影响,将本文采用的 MRWR 算法和 MRW 算法做对比,迭代次数分别设置为 10 和 100,每次实验随机重复生成查询节点 10 次,对 10 次结果的 ANC 值取平均处理,查询节点个数对比实验结果如图 8 所示。

从图 8 来看,随着查询节点个数的增加,局部拓扑结构的连通性逐步下降。这与理论分析是一致的,假设查询节点局部相邻,查询节点数目增多会使得游走到与查询节点距离较远的节点的概率增大;假设查询节点局部不相邻,会减弱局

部拓扑结构中节点的聚集度,从而导致 ANC 值下降。同时,MRWR 的 ANC 比 MRW 更高,这是因为 MRWR 的重启策略将节点访问概率限制在查询节点周围,且同一算法迭代次数越高,结果越稳定。综上考虑,每次输入的查询节点个数不超过 6 为最优。

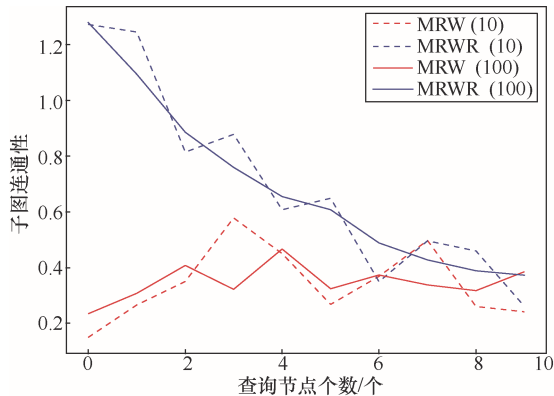


图 8 查询节点个数对比实验结果

Figure 8 The experimental results of the comparison of the number of query nodes

3.3.3 不同随机游走算法实验分析

(1) 算法性能对比分析

实验的参数设置如下: 查询节点个数为 2 和 3, 重启概率=0.15, 迭代次数为 10; $K=10$ 。在实验过程中取每个算法生成的 top 10 节点, 并计算 top 10 节点的准确率。

不同算法生成 top10 结果的准确率如表 2 所示, 计算结果表明, 在多个数据集上, 本文提出的 MQ-KNM 算法准确率比其他 5 种对比算法高, 能够较好地挖掘查询节点相关的关键节点。

表 2 不同算法生成 top10 结果的准确率
Table 2 The accuracy of top10 results of different algorithms

数据集	DC	SOC	BC	VR	EC	MQ-KNM
cit-DBLP-2	0.5	0.7	0.4	0.2	0.4	0.8
cit-DBLP-3	0.6	0.7	0.5	0.3	0.5	1
email-EU-2	0.8	0.5	0.8	0.5	0.8	0.8
email-EU-3	0.5	1	0.2	/	0.5	0.6

(2) 可视化实例分析

本文提出的 MQ-KNM 算法在 DBLP 数据集上分别选取 1 个、2 个、3 个查询节点生成 10 个

关键节点, 图 9~图 11 为结果可视化。

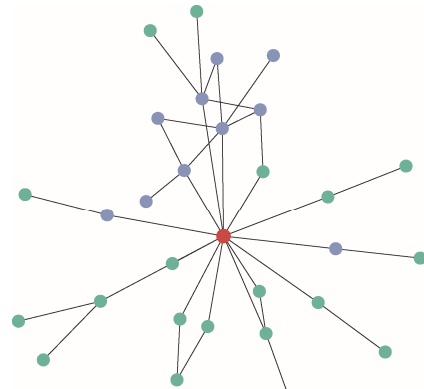


图 9 1 个查询节点的结果可视化
Figure 9 Result visualization of a query node

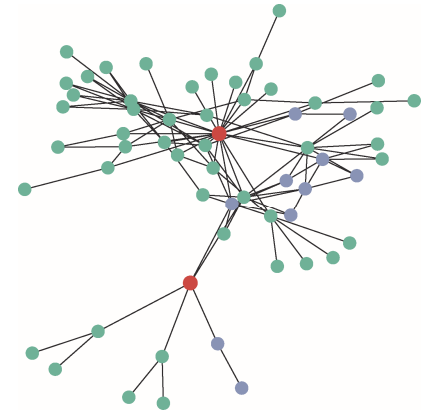


图 10 2 个查询节点的结果可视化
Figure 10 Result visualization of two query nodes

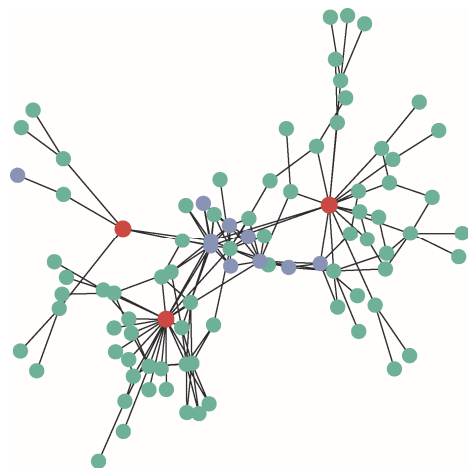


图 11 3 个查询节点的结果可视化
Figure 11 Result visualization of three query nodes

图 9~图 11 中红色节点表示查询节点, 绿色节点表示普通节点, 蓝色节点表示挖掘出的前 10

个关键节点。从图中可以看出, 关键节点大部分集中于查询节点所在局部拓扑的中心位置, 直观说明本文算法能够较好地找出局部拓扑结构与查询节点相关的关键节点。

综上所述, 第一阶段的 LTS 局部拓扑结构生成算法运行时间较短, 可以有效减小第二阶段的计算规模; 从挖掘出的关键节点的准确率和对实验结果的可视化展示中可以看出, 本文 MQ-KNM 算法可以有效挖掘查询节点相关的关键节点。

4 结束语

本文基于随机游走算法和 `graphsage` 模型提出一种两阶段的基于查询节点的关键节点挖掘算法。该算法首先生成与多个查询节点相关的局部拓扑结构, 然后学习局部拓扑结构中节点的嵌入向量, 最终通过嵌入向量的相似度来量化节点的关键程度, 得到关键节点列表。通过对实验结果的分析, 本文所提算法框架与其他典型算法相比, 具有多种优点。需要指出的是, 目前本文使用随机游走算法生成的局部拓扑结果存在随机性, 最终生成的局部拓扑结构不够稳定。本文算法在参数优化和算法时间空间复杂度方面还有进一步提升空间。

参考文献:

- [1] 任晓龙, 吕琳媛. 网络重要节点排序方法综述[J]. 科学通报, 2014, 59(13): 1175-1197.
- REN X L, LYU L Y. Review of ranking nodes complex networks[J]. Science Bulletin, 2014, 59(13): 1175-1197.
- [2] LU P, DONG C. Ranking the spreading influence of nodes in complex networks based on mixing degree centrality and local structure[J]. International Journal of Modern Physics B, 2019, 33(32): 1950395.
- [3] FAN Y, ZHANG R, ZHAO Y, et al. Identifying the most influential spreaders in complex networks by an extended local K-shell sum[J]. International Journal of Modern Physics C, 2017, 28(1): 925-214.
- [4] SARA A, HASSAN B. Identification of influential spreaders in complex networks using HybridRank algorithm[J]. Scientific Reports, 2018, 8(1): 11932.
- [5] HAI J, QIAN C, YU D, et al. Parallel computation of hierarchical closeness centrality and applications[C]//Proceedings of World Wide Web. 2018: 3047-3064.
- [6] NEWMAN M E J. A measure of betweenness centrality based on random walks[J]. Social Networks, 2004, 27(1): 39-54.
- [7] BRIN, S, PAGE L, MOTWAMI R, et al. The PageRank citation ranking: bringing order to the Web[Z]. Stanford Digital Libraries Working Paper, 1998(6): 102-107.
- [8] XU S, WANG P. Identifying important nodes by adaptive Leader-Rank[J]. Physica A: Statistical Mechanics and its Applications, 2017, 469: 654-664.
- [9] 陈勇, 胡爱群, 胡啸. 通信网中节点重要性的评价方法[J]. 通信学报, 2004, 25(8): 129-134.
- CHEN Y, HU A Q, HU X. Evaluation method of node importance in communication network[J]. Journal on Communications, 2004, 25(8): 129-134.
- [10] 谭跃进, 吴俊, 邓宏钟. 复杂网络中节点重要度评估的节点收缩方法[J]. 系统工程理论与实践, 2006, (11): 79-83.
- TAN Y J, WU J, DENG H Z. Node importance in complex networks[J]. Systems Engineering Theory and Practice, 2006, (11): 79-83.
- [11] RASHID Y, BHAT J I. OlapGN: a multi-layered graph convolution network-based model for locating influential nodes in graph networks[J]. Knowledge-Based Systems, 2024, (283).
- [12] BOUYER A, BENI H A, ARASTEH B, et al. FIP: A fast overlapping community-based influence maximization algorithm using probability coefficient of global diffusion in social networks[J]. Expert Systems with Applications, 2023, (213).
- [13] AGHAALIZADEH S, AFSHORD S T, BOUYER A, et al. A three-stage algorithm for local community detection based on the high node importance ranking in social networks[J]. Physica A: Statistical Mechanics and its Applications, 2021, 63(25420).
- [14] WATTS D J, STROGATZ S H. Collective dynamics of small world network[J]. Nature, 1998, 393(4): 440-442.
- [15] BAHADORI S, MORADI P, ZARE H. An improved limited random walk approach for identification of overlapping communities in complex networks[J]. Applied Intelligence, 2020, 51: 3561-3580.
- [16] LIU J, ONG G P, CHEN X M. GraphSAGE-based traffic speed forecasting for segment network with sparse data[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(3): 1755-1766.
- [17] YAN Y, LUO D, NI J, et al. Local graph clustering by multi-network random walk with restart[C]//Proceedings of Pacific-asia Conference on Knowledge Discovery & Data Mining. 2018: 490-501.
- [18] SEIDMAN S B. Network structure and minimum degree[J]. Social Networks, 1983, 5(3): 269-287.
- [19] 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, 2010: 939-948.
- [20] ROSSI R A, AHMED N K. An interactive data repository with visual analytics[J]. ACM SIGKDD Explorations Newsletter, 2016, 17(2): 37-41.
- [21] ARASTEH M, ALIZADEH S. A fast divisive community detection algorithm based on edge degree betweenness centrality[J]. Applied Intelligence, 2019, 49(2): 689-702.
- [22] KERMARREC A M, MERRER E L, SERICOLA B, et al. Second order centrality: Distributed assessment of nodes criticality in complex networks[J]. Computer Communications, 2010, 34(5): 619-628.

- [23] TSALOUCHIDOU I, BAEZA-YATES R, BONCHI F, et al. Temporal betweenness centrality in dynamic graphs[J]. International Journal of Data Science and Analytics, 2020, 9(3): 257-272.
- [24] ZHANG J X, CHEN D B, DONG Q, et al. Identifying a set of influential spreaders in complex networks [J]. Scientific Reports, 2016, 6(1): 27823.
- [25] SONG YANG. Networks: an Introduction by M. E. J. Newman[J]. The Journal of Mathematical Sociology, 2013, 37(4): 250-251.
- [26] BEINEKE L W, OELLERMANN O R, PIPPERT R E. The average connectivity of a graph[J]. Discrete Mathematics, 2002, 252(1-3): 31-45.
- [27] LUO D S, BIAN Y C, YAN Y W, et al. Local community detection in multiple networks[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, 2020: 266-274.



黄俊恒（1966—），男，河南新乡人，哈尔滨工业大学（威海）副教授，主要研究方向为社交网络、数据挖掘。



魏家扬（2000—），男，江西南昌人，哈尔滨工业大学（威海）硕士生，主要研究方向为复杂网络分析、GNN。

[作者简介]



辛国栋（1976—），男，山东青岛人，哈尔滨工业大学（威海）讲师，主要研究方向为网络信息挖掘、网络安全、机器学习。



刘润莹（2003—），男，山东德州人，主要研究方向为社交网络和自然语言处理。



朱滕威（1998—），男，山东枣庄人，哈尔滨工业大学（威海）工程师，主要研究方向为复杂网络分析。



王巍（1975—），女，黑龙江齐齐哈尔人，哈尔滨工业大学（威海）副教授，主要研究方向为社交网络，数据挖掘和自然语言处理。