



计算机科学

COMPUTER SCIENCE

融合动态距离和随机竞争学习的社区发现算法

王本钰, 顾益军, 彭舒凡, 郑棣文

引用本文

王本钰, 顾益军, 彭舒凡, 郑棣文. [融合动态距离和随机竞争学习的社区发现算法](#)[J]. 计算机科学, 2022, 49(5): 170-178.

WANG Ben-yu, GU Yi-jun, PENG Shu-fan, ZHENG Di-wen. [Community Detection Algorithm Based on Dynamic Distance and Stochastic Competitive Learning](#)[J]. Computer Science, 2022, 49(5): 170-178.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[面向双层网络的 EWCC 社区发现算法](#)

EWCC Community Discovery Algorithm for Two-Layer Network

计算机科学, 2022, 49(4): 49-55. <https://doi.org/10.11896/jsjcx.210800275>

[基于结构深度网络嵌入模型的节点标签分类算法](#)

Node Label Classification Algorithm Based on Structural Depth Network Embedding Model

计算机科学, 2022, 49(3): 105-112. <https://doi.org/10.11896/jsjcx.201000177>

[基于节点相似性和网络嵌入的复杂网络社区发现算法](#)

Complex Network Community Detection Algorithm Based on Node Similarity and Network Embedding

计算机科学, 2022, 49(3): 121-128. <https://doi.org/10.11896/jsjcx.210200009>

[于路径连接强度的有向网络链路预测方法](#)

Link Prediction Method for Directed Networks Based on Path Connection Strength

计算机科学, 2022, 49(2): 216-222. <https://doi.org/10.11896/jsjcx.210100107>

[基于信息传播的致病基因识别研究](#)

Disease Genes Recognition Based on Information Propagation

计算机科学, 2022, 49(1): 264-270. <https://doi.org/10.11896/jsjcx.201100129>

融合动态距离和随机竞争学习的社区发现算法

王本钰 顾益军 彭舒凡 郑棣文

中国人民公安大学信息安全学院 北京 100032

(201621430015@stu.ppsuc.edu.cn)

摘要 社区结构作为复杂网络的一个重要属性,对于了解复杂网络的组织架构和功能具有深远意义。为了解决复杂网络的社区发现问题,提出了一种融合动态距离和随机竞争学习的社区发现算法(Dynamic Distance Stochastic Competitive Learning, DDSCL)。该算法首先结合节点度值和节点间的欧氏距离来确定随机竞争学习中粒子的初始位置,使得不同粒子在游走初期不会在同一社区内进行竞争,加快了粒子的收敛速度;然后结合动态距离算法,将节点间的动态距离融入粒子优先游走过程中,使得粒子的优先游走过程更具方向性,减小了随机性,并且粒子游走的过程也会优化动态距离的变化;当粒子达到收敛状态时,节点将被对其具有最大控制力的粒子占据。最后网络中每一个粒子对应一个社区,根据各粒子占据的节点来揭示网络的社区结构。在8个真实的网络数据集上,以NMI和模块度 Q 值为评价指标,将DDSCL算法与现有的代表性算法进行实验比较,发现DDSCL算法整体上优于其他算法,其不仅降低了随机竞争学习中粒子优先游走的随机性,而且解决了动态距离算法中出现的碎片化社区问题,提高了社区发现结果的准确性。实验结果表明,所提算法具有有效性和可行性。

关键词: 复杂网络; 社区发现; 动态距离; 随机竞争学习; 粒子竞争

中图分类号 TP391

Community Detection Algorithm Based on Dynamic Distance and Stochastic Competitive Learning

WANG Ben-yu, GU Yi-jun, PENG Shu-fan and ZHENG Di-wen

School of Information and Network Security, People's Public Security University of China, Beijing 100032, China

Abstract Community structure is an important property of complex networks. It is profoundly significant for understanding the organizational structure and functions of complex networks. A community detection algorithm (Dynamic Distance Stochastic Competitive Learning, DDSCL) is proposed to solve the community detection problem of complex networks. DDSCL is based on dynamic distance and stochastic competitive learning. The algorithm first combines node degree values and Euclidean distances between nodes to determine the initial positions of particles in stochastic competitive learning, which will allow different particles to not compete within the same community at the beginning of the wander, speeding up the convergence of the particles. The dynamic distance between nodes is then combined with a dynamic distance algorithm to incorporate the dynamic distance between nodes into the particle prioritization walking process. The particle prioritization process is more directional and less random in this way. The particle travel process will also optimize the change in dynamic distance. When the particles reach a convergence state, the node is occupied by the particle that has the most control over it. Each particle in the network eventually corresponds to a community, and the community structure of the network is revealed according to the nodes occupied by each particle. DDSCL is compared in experimental tests on eight real network datasets, and it uses NMI and modularity Q -value as evaluation metrics. It's found that DDSCL outperforms other algorithms overall. The algorithm first reduces the randomness of preferential walking of particles in stochastic competitive learning. Then DDSCL solves the problem of fragmented communities arising from dynamic distance algorithms, and improves the accuracy of community detection results. The experimental results show the proposed algorithm's effectiveness.

Keywords Complex network, Community detection, Dynamic distance, Stochastic competitive learning, Particle competition

到稿日期:2021-03-21 返修日期:2021-05-18

基金项目:公安部科技强警基础工作专项项目(2020GABJC02);中国人民公安大学基本科研业务费项目(2021JKF420)

This work was supported by the Ministry of Public Security Science and Technology to Strengthen the Basic Work of the Police Project (2020GABJC02) and Basic Research Business Fee Project of People's Public Security University of China(2021JKF420).

通信作者:顾益军(guyijun@ppsuc.edu.cn)

1 引言

随着社会信息化技术的发展,事物之间的相互关系大多可以通过复杂网络的形式呈现出来,如社交应用中的好友关系、生物蛋白质中的 PPI 网络、Internet 中网页链接关系以及论文中的相互引用关系^[1]。在此类网络中,常常可以把单个个体抽象为节点,个体与个体之间的联系抽象为边,社区发现的任务就是找到网络中联系紧密的节点^[2]。社区发现是复杂网络的一个热点研究方向,社区内的节点往往比网络社区外的节点具有更紧密的联系^[3]。属于同一社区的节点可能具有更高的相关性,找到同一社区内的节点对于了解网络的组织架构和功能具有重要意义。

基于网络动力学的方法是社区发现算法的一个重要分支,目前主流的基于网络动力学的社区发现算法有马尔可夫聚类^[4]、随机游走^[5]、标签传播^[6]、随机竞争学习^[7]、动态距离^[8]等。Wang 等^[4]提出基于改进马尔可夫聚类的社区发现算法,改进了随机流和转移概率间的迭代过程,在静态和动态网络中都取得了不错的效果。Bian 等^[5]提出 RWM 算法,将随机游走算法应用于多层网络社区发现任务中。EI 等^[6]提出 NI-LPA 算法,利用节点重要性改进了标签传播算法中标签传播和过滤的过程,在重叠网络社区发现任务上取得了较好的效果。Silva 等^[7]提出随机竞争学习,该方法以粒子竞争机制为基础,以无监督学习的方式来实现网络的社区发现任务。Shao 等^[8]提出了基于节点间动态距离的社区发现方法,该方法认为网络中节点间的距离会随时间动态变化并最终达到收敛状态,实验结果证明该方法取得了很好的效果。

竞争性学习是一种重要的机器学习方法,已被广泛应用于人工神经网络以实现无监督学习。早期的发展包括著名的自组织图^[9]、差分竞争学习^[10]和自适应共振理论^[11]。基于粒子竞争机制的社区发现方法最初由 Quiles 等^[12]提出,他们研究了复杂网络中粒子的随机性和确定性,粒子通过既定的规则在网络中随机游走并且互相竞争,通过粒子占有更多节点的方式来达到社区发现的效果。Silva 等^[7]完善了粒子竞争算法,提出了随机竞争学习,以无监督学习的方式来实现社区发现任务,在复杂网络系统动力学领域具有推动作用。以随机竞争学习为基础,用于检测和防止错误标签的方法被提出,该方法通过同一类的粒子在网络中互相合作、不同类的粒子在网络中相互竞争的形式来防止错误标签在网络中的传播^[13]。此后,在半监督社区发现^[14]、异常节点检测^[15]、多层网络社区发现^[16]、分类数量预测^[17]以及图像分割^[18]问题的解决中纷纷引入了随机竞争学习。随机竞争学习的发展历程证明了其应用于复杂网络社区发现任务的可行性和有效性。

Shao 等^[8]提出的基于节点间动态距离的社区发现方法对于复杂网络动力学的发展也有深远意义。Li 等^[19]将动态距离算法扩展至多层网络划分中,以时间为度量逐渐加强了网络内连接和网络间耦合的强度。对于两个网络中的节点,考虑了与此节点关联的网络内的连接以及与其邻居节点相关联的网络间的耦合,以更新多个网络间耦合的强度。Sun 等^[20]将动态距离算法从单分网络扩展到了二分网络,扩展了在二分网络中计算 Jaccard 距离的方法。Sun 等^[21]提出了

一种基于动态距离的重叠语义社区发现算法,该算法能够检测出结构紧凑、属性同质的高质量语义社区。Wan 等^[22]提出了一种改进的动态距离社区发现算法,该算法的核心思想是当节点间的动态距离变化出现单调性趋势时,即认为节点间动态距离总是连续增加或者减少,解决了动态距离算法中更新过程收敛速度慢的问题。在近几年社区发现算法的发展过程中,动态距离算法已经被广泛应用于社区发现任务中。

本文通过研究发现,随机竞争学习中的粒子竞争机制在粒子初期的优先游走过程具有很大的随机性并且粒子收敛速度慢,导致社区发现的结果不准确。动态距离算法会出现碎片化社区问题,导致社区发现的准确性差。因此,本文结合动态距离和随机竞争学习算法,提出了一种新的社区发现算法(Dynamic Distance Stochastic Competitive Learning, DDSCL)。本文的主要贡献有以下几个方面:

(1)结合节点度值和节点间的欧氏距离确定随机竞争学习中粒子的初始位置,使得不同粒子在游走初期不会在同一社区内进行竞争,以加快粒子的收敛速度。

(2)将动态距离算法引入随机竞争学习算法中,使得粒子的优先游走过程更具方向性,减小了随机性。

(3)将随机竞争学习中粒子游走的过程用于优化动态距离的变化,使动态距离的计算更加准确。

(4)DDSCL 算法不仅降低了随机竞争学习中粒子优先游走的随机性,而且解决了动态距离算法中出现的碎片化社区问题,提高了社区发现结果的准确性。在公开的 8 个数据集上对 DDSCL 算法进行实验测试比较,结果表明 DDSCL 算法社区发现的准确性具有显著的提升,同时对主要参数进行了讨论。

2 相关工作

2.1 网络基础知识

给定复杂网络 $G=(V, E)$, 其中, $V=\{v_1, v_2, \dots, v_n\}$ 为网络中个体的集合, 且 $|V(G)|=n$, 边集 $\{1 \leq i, j \leq n\}$ 为网络个体间联系的集合。本文仅对无向简单图进行讨论, 将 $e=(v_i, v_j)$ 记为 $v_i v_j$ 。

记网络 G 的邻接矩阵 $A \in R^{N \times N}$, 具体定义为:

$$a_{ij} = \begin{cases} 1, & v_i v_j \in E(G) \\ 0, & v_i v_j \notin E(G) \end{cases} \quad (1)$$

令 $Non(v_i) = \{v_j | v_j \in V \wedge (v_i, v_j) \in E\}$, 称 $Non(v_i)$ 为节点 v_i 的邻居节点组成的集合。

在网络 G 中, 节点 $E = \{e = (v_i, v_j)\}$, v_i 和 v_j 间的 Jaccard 距离定义为^[23]:

$$Jac(v_i, v_j) = 1 - \frac{|Non(v_i) \cap Non(v_j)|}{|Non(v_i) \cup Non(v_j)|} \quad (2)$$

Jaccard 距离能计算节点间的局部距离, 其值越小, 说明两个节点共有的邻居节点就越多, 节点的相似度也就越高。

2.2 随机竞争学习

Silva 等^[7]提出了随机竞争学习, 随机竞争学习是一种依据粒子间竞争的动力学算法, 通过无监督的方式来实现复杂网络上的社区发现任务。给定复杂网络 $G=(V, E)$, V 是网络中节点的集合, E 是网络中边的集合。在随机竞争学习中,

以随机的方式将多个粒子置于网络中,每个粒子在网络中按照既定的规则进行游走,粒子尽可能地占领更多的节点以扩充自己的领地,同时保护已经被自己占据的节点不受其他粒子的控制。通过这种方式产生粒子间的竞争,当粒子访问节点时,会加强粒子对该节点的控制能力,同时会削弱其他粒子对这个节点的控制能力。随机竞争学习中每个粒子对应一个社区,以粒子达到收敛状态时每个粒子占据的节点来揭示网络的社区结构。

在随机竞争学习中,粒子的状态主要由 $P(t)$, $N(t)$, $E(t)$, $S(t)$ 这 4 个变量进行控制。 $P(t)$ 表示在 t 时刻粒子在网络中所处的位置; $N(t)$ 表示粒子对于节点的访问次数,当粒子访问节点次数较多时,对于该节点则具有更强的控制能力; $E(t)$ 表示粒子的能级,当粒子访问别的粒子占据的节点时,粒子的能级会降低,当粒子访问自身占据的节点时,能级会提高,粒子能级的变化值用 Δ 表示; $S(t)$ 表示粒子当前的能量状态,当粒子能量耗尽时,粒子将会进入休眠状态,该粒子将被重置,且会在网络中被重新生成。

在粒子的游走过程中,粒子以随机游走和优先游走相结合的规则进行游走。随机游走体现了粒子的随机性,粒子将以相同的概率在邻居节点上进行游走。而优先游走则体现了粒子的倾向性,对于自己已经占领的节点将会以更高的概率进行访问。粒子的随机游走和优先游走的定义如下:

$$P_{\text{rand}}^{(k)}(i, j) \triangleq \frac{a_{i,j}}{\sum_{u=0}^{|V|} a_{i,u}} \quad (3)$$

$$P_{\text{pref}}^{(k)}(i, j, t) \triangleq \frac{a_{i,j} \bar{N}_j^{(k)}(t)}{\sum_{u=0}^{|V|} a_{i,u} \bar{N}_u^{(k)}(t)} \quad (4)$$

其中, $\bar{N}_j^{(k)}(t)$ 表示在 t 时刻 k 个粒子访问节点 v_j 的次数的比, $N_j(t) \triangleq [N_j^1(t), N_j^2(t), \dots, N_j^k(t)]$ 表示网络中 k 个粒子访问节点 v_j 的次数。因此,粒子对节点控制能力的定义如下:

$$\bar{N}_j^k(t) \triangleq \frac{N_j^k(t)}{\sum_{u=1}^K N_j^u(t)} \quad (5)$$

$$\bar{N}_j(t) \triangleq [\bar{N}_j^1(t), \bar{N}_j^2(t), \dots, \bar{N}_j^k(t)] \quad (6)$$

其中, $\bar{N}_j^k(t)$ 表示在 t 时刻粒子 k 对于节点 v_j 的控制力,粒子对于节点的控制能力由粒子访问节点的次数决定; $\bar{N}_j(t)$ 表示网络中 k 个粒子对节点 v_j 的控制力大小,称作节点粒子控制力向量。

随机竞争学习通过参数 λ 来调节随机游走和优先游走发生的概率,当 λ 为 1 时,粒子在网络中只进行优先游走;当 λ 为 0 时,粒子在网络中只进行随机游走;当 λ 取值为 $(0, 1)$ 时,粒子在网络中同时进行随机游走和优先游走。因此,粒子在网络中的游走公式的定义如下:

$$P^{(k)}(t) \triangleq \lambda P_{\text{pref}}^{(k)}(t) + (1-\lambda) P_{\text{rand}}^{(k)}(t) \quad (7)$$

当粒子在游走过程中,达到收敛状态时,粒子将不再游走,收敛状态公式的定义如下:

$$\|\bar{N}(t) - \bar{N}(t-1)\|_{\infty} < \epsilon \quad (8)$$

其中, ϵ 为收敛阈值,一般取值为 0.05。

粒子结束游走后,网络中的节点将被对其具有最大控制力的粒子占据。最后,网络中的每一个粒子对应一个社区,根据每个粒子占据节点的情况来揭示网络中的社区结构。

综上,随机竞争学习可以根据式(9)来进行描述。

$$X(t) = \begin{bmatrix} P(t) \\ N(t) \\ E(t) \\ S(t) \end{bmatrix} \quad (9)$$

2.3 动态距离算法

Shao 等^[8]提出了动态距离算法,动态距离算法基于复杂网络动力学,其核心理论为:随着时间的推移,节点间的间距是动态变化的。两个节点间的距离会受 3 种交互模式的影响而进行动态变化,3 种影响分别是直接连接节点影响、共同邻居节点影响和独立邻居节点影响。节点间的距离会随着时间的推移逐渐增大或者减小至一个稳定值,直至整个复杂网络达到一个稳定的状态,使得网络的结构显现出来。动态距离算法的 3 种交互模式的影响如图 1 所示。

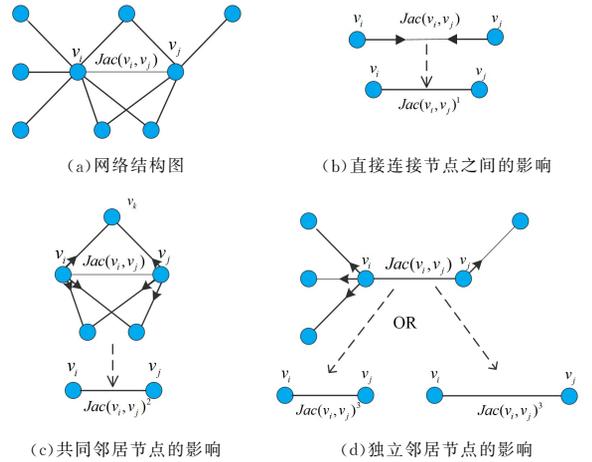


图 1 动态距离的 3 种交互模式影响

Fig. 1 Three interaction modes of dynamic distance

交互模式 1 直接连接节点之间的影响。直接连接的两个节点会相互吸引,使得两个节点之间的距离逐渐变小。这种直接连接节点的交互定义为:

$$DNI(v_i, v_j) = - \left(\frac{f(1 - Jac(v_i, v_j))}{K_i} + \frac{f(1 - Jac(v_i, v_j))}{K_j} \right) \quad (10)$$

其中, $f(\cdot)$ 是形为 $\sin(\cdot)$ 的耦合函数, $Jac(v_i, v_j)$ 是节点 v_i 和 v_j 的 Jaccard 距离, K_i 和 K_j 是节点 v_i 和 v_j 的度值, $\frac{1}{K_i}$ 和 $\frac{1}{K_j}$ 为节点 v_i 和 v_j 的归一化因子。节点 v_i 和 v_j 间的 Jaccard 距离越小,直接连接节点间的影响就越大。

交互模式 2 共同邻居节点的影响。其中共同邻居节点用 CN 表示,定义 $CN(v_i, v_j) = \{Non(v_i) - v_j\} \cap \{Non(v_j) - v_i\}$ 。节点 v_i 和 v_j 间的共同邻居节点会吸引两个节点靠近,使得两个节点的动态距离变小。这种共同邻居节点的交互定义为:

$$CNI(v_i, v_j) = - \sum_{v_k \in CN} \left(\frac{1}{K_i} \cdot f(1 - Jac(v_k, v_i)) \cdot (1 - Jac(v_k, v_i)) + \frac{1}{K_j} \cdot f(1 - Jac(v_k, v_j)) \cdot (1 - Jac(v_k, v_j)) \right) \quad (11)$$

交互模式 3 独立邻居节点的影响。节点的独立邻居

节点用 EN 表示,节点 v_i 相对于节点 v_j 的独立邻居节点定义为: $EN(v_i) = \{Non(v_i) - \{Non(v_i) \cap Non(v_j)\}\}$; 节点 v_j 相对于节点 v_i 的独立邻居定义为: $EN(v_j) = \{Non(v_j) - \{Non(v_i) \cap Non(v_j)\}\}$ 。为了衡量两个节点间独立邻居节点对节点间距离的影响,引入参数 ϕ ,影响程度用函数 γ 表示,定义如下:

$$\gamma(v_k, v_i) = \begin{cases} 1 - Jac(v_k, v_j), & 1 - Jac(v_k, v_j) \geq \phi \\ 1 - Jac(v_k, v_j) - \phi, & \text{otherwise} \end{cases} \quad (12)$$

其中, $v_k \in EN(v_i)$ 。独立邻居节点的交互定义为:

$$ENI(v_i, v_j) = - \sum_{v_m \in EN(v_i)} \left(\frac{1}{K_i} \cdot f(1 - Jac(v_m, v_i)) \cdot \beta(v_m, v_i) \right) - \sum_{v_n \in EN(v_j)} \left(\frac{1}{K_j} \cdot f(1 - Jac(v_n, v_j)) \cdot \beta(v_n, v_j) \right) \quad (13)$$

结合上述 3 种交互模式,当时间为 t 时,节点 v_i 和 v_j 间的动态距离变化定义为:

$$Jac(v_i, v_j, t+1) = Jac(v_i, v_j, t) + DNI(v_i, v_j, t) + CNI(v_i, v_j, t) + ENI(v_i, v_j, t) \quad (14)$$

3 DDSCL 算法

为了准确、合理地挖掘复杂网络社区结构,并考虑到复杂网络是一个动力学系统,我们引入动态距离算法和随机竞争学习相结合的 DDSCL 算法来执行社区发现任务。

3.1 粒子初始位置的确定

在随机竞争学习中,粒子的初始位置是被随机选择的。这种粒子随机选择的方式可能会导致不同粒子的初始位置在相同的社区内,使得粒子游走初期在同一社区内不断竞争。虽然粒子最终会在不同社区内达到收敛状态,但是粒子游走的时间会增加,导致粒子收敛的速度变慢。因此,本文结合节点度值和节点间欧氏距离来确定粒子的初始位置,使得粒子的初始位置在不同社区内,让粒子游走初期不会在同一社区内开展竞争,加快粒子的收敛速度。

欧氏距离侧重考虑的是节点间的全局距离,因此本文将节点 v_i 和 v_j 表示为由 Jaccard 距离扩展而成的向量 $\mathbf{T}(v_i)$ 和向量 $\mathbf{T}(v_j)$,再计算向量间的欧氏距离^[24],以弥补 Jaccard 距离仅考虑节点间局部距离这一缺陷。

向量 $\mathbf{T}(v_i)$ 和向量 $\mathbf{T}(v_j)$ 由 Jaccard 距离扩展而成,即 $\mathbf{T}(v_i) = [Jac(v_i, v_1), \dots, Jac(v_i, v_2), \dots, Jac(v_i, v_n)]$, $\mathbf{T}(v_j) = [Jac(v_j, v_1), \dots, Jac(v_j, v_2), \dots, Jac(v_j, v_n)]$ 。向量 $\mathbf{T}(v_i)$ 和向量 $\mathbf{T}(v_j)$ 的欧氏距离 $dist(v_i, v_j)$ 如式(15)所示:

$$dist(v_i, v_j) = \sqrt{\sum_{k=1}^n (Jac(v_i, v_k) - Jac(v_j, v_k))^2} \quad (15)$$

设粒子初始位置的集合为 $V_{ini} = \{v_1, v_2, \dots, v_k\}$, k 为粒子的数量。在确定粒子初始位置时,首先将网络中的节点按度值进行排序,选取度值最大的节点作为第一个粒子的初始位置。接着,将与网络中已有粒子所在节点的欧氏距离平均值最大的节点作为下一个粒子的初始位置。最后,当粒子数量为 k 时,完成粒子位置初始任务。粒子初始位置如式(16)所示:

$$v_k = \begin{cases} v_i, & \text{if } k=1 \text{ and } K_i = \max_j(K_j) \\ v_j, & \text{if } k \geq 2 \text{ and } \max_j \left(\frac{\sum_{n=1}^k dist(v_n, v_j)}{k} \right) \end{cases} \quad (16)$$

3.2 结合随机竞争学习的动态距离优化方式

在动态距离算法中,节点之间的距离会随着时间的变化而变化。随机竞争学习也是一种网络动力学算法,在粒子游走过程中,粒子对于节点的控制力也会随着时间的变化而变化。复杂网络是一个大型动力学系统,因此粒子游走的动态过程也会对节点间的距离产生影响,可以用来优化动态距离的变化。粒子对于节点控制力的变化揭示了粒子游走的过程,因此本文将体现粒子对于节点控制力变化的节点粒子控制力向量作为一种交互模式,用于优化节点间的动态距离。

首先通过广义的 Jaccard 相似性^[25]来计算节点粒子控制力向量的相似度。节点 v_i 和 v_j 的粒子控制力向量的广义 Jaccard 相似性如式(17)所示:

$$EJ(i, j) = \frac{\bar{\mathbf{N}}_i(t) \cdot \bar{\mathbf{N}}_j(t)}{\|\bar{\mathbf{N}}_i(t)\|^2 + \|\bar{\mathbf{N}}_j(t)\|^2 - \bar{\mathbf{N}}_i(t) \cdot \bar{\mathbf{N}}_j(t)} \quad (17)$$

其中, $\bar{\mathbf{N}}_i(t)$ 和 $\bar{\mathbf{N}}_j(t)$ 分别表示节点 v_i 和 v_j 的粒子控制力向量, $\bar{\mathbf{N}}_i(t) \triangleq [\bar{N}_i^1(t), \bar{N}_i^2(t), \dots, \bar{N}_i^k(t)]$, $\bar{\mathbf{N}}_j(t) \triangleq [\bar{N}_j^1(t), \bar{N}_j^2(t), \dots, \bar{N}_j^k(t)]$ 。Jaccard 相似性计算的向量元素只可以为 0 或者 1,而节点粒子控制力向量元素取值在 $(0, 1)$ 之间,因此 Jaccard 相似性不适用于计算节点粒子控制力向量的相似性。而广义 Jaccard 相似性计算向量元素的取值可以在 $[0, 1]$ 之间,因此本文使用广义 Jaccard 相似性来计算节点粒子控制力向量的相似度。

广义 Jaccard 距离由其相似性计算而来,节点 v_i 和 v_j 的粒子控制力向量的广义 Jaccard 距离如式(18)所示:

$$h(i, j) = 1 - EJ(i, j) \quad (18)$$

当两个节点的粒子控制力向量相似度较高时,即网络中的粒子对于这两个节点的控制力大体相同,根据随机竞争学习理论,两个节点归属于同一社区的可能性更大,两个节点的实际距离也应该更小。为了刻画节点粒子控制力向量对于节点间距离的影响程度,引入参数 δ ,节点 v_i 和 v_j 的粒子控制力向量的广义 Jaccard 距离 $h(i, j)$ 对于节点距离的影响程度 $\beta(i, j)$ 如式(19)所示:

$$\beta(i, j) = \begin{cases} 1 - h(i, j), & 1 - h(i, j) \geq \delta \\ 1 - h(i, j) - \delta, & \text{otherwise} \end{cases} \quad (19)$$

当两个节点间的粒子控制力向量相似性大于参数 δ 时,影响程度 $\beta(i, j)$ 为正值,节点间的距离会减小;当两个节点间的粒子控制力向量相似性小于参数 δ 时,影响程度 $\beta(i, j)$ 为负值,节点间的距离会增加。因此,参照动态距离算法,节点粒子控制力向量交互 NDI 如式(20)所示:

$$NDI(i, j) = - \left(\frac{f(1 - h(i, j)) \cdot \beta(i, j)}{K_i} + \frac{f(1 - h(i, j)) \cdot \beta(i, j)}{K_j} \right) \quad (20)$$

结合动态距离算法的 3 种交互模式和本文提出的节点粒子控制力向量交互模式,节点 v_i 和 v_j 间的距离 $Jac(i, j)$ 随时间变化而更新的公式如式(21)所示:

$$Jac(i, j, t+1) = Jac(i, j, t) + DNI(i, j, t) + CNI(i, j, t) + ENI(i, j, t) + NDI(i, j, t) \quad (21)$$

3.3 结合动态距离的随机竞争学习粒子优先游走方式

随机竞争学习中,粒子的游走概率由优先游走概率和随机游走概率组成。然而,粒子在初期游走中,由于粒子并没有建立优先游走的倾向性,使得粒子的优先游走过程没有方向性,具有很大的随机性,这样会导致粒子优先游走概率受粒子初期游走过程影响大,导致社区发现的准确率降低。因此,本文将动态距离算法应用于随机竞争学习中,将节点间的动态距离变化告知粒子,使得粒子进行优先游走时,以更高的概率向距离小的节点游走。对于距离大的节点,粒子则会以更小的概率向该节点进行游走。结合动态距离的随机竞争学习,使得粒子优先游走过程具有更好的方向性和确定性,可以更好地揭示网络的社区结构,而粒子游走的随机性则更多体现在随机游走过程中。因此,粒子的优先游走概率公式如式(22)所示:

$$P_{\text{pref}}(i, j, t) \triangleq \frac{a_{ij} \bar{N}_j^{(k)}(t) \frac{1}{(1+Jac(i, j, t))^2}}{\sum_{u=1}^V a_{iu} \bar{N}_u^{(k)}(t) \frac{1}{(1+Jac(i, j, t))^2}} \quad (22)$$

其中, a_{ij} 表示节点 v_i 和 v_j 之间是否具有连接, $\bar{N}_j^{(k)}$ 表示粒子 k 对于节点 v_j 的控制能力, $Jac(i, j, t)$ 表示节点 v_i 和 v_j 之间的动态距离。

融合动态距离的随机竞争学习中,动态距离是作为粒子优先游走概率的一个计算指标,因此动态距离的更新只需要根据网络的规模在粒子游走过程中进行少量迭代,不同于动态距离算法让每个节点达到收敛状态,即节点到任意节点的距离为 0 或 1。融合动态距离的随机竞争学习,通过减少动态距离的迭代次数,可以降低动态距离算法的时间复杂度,提高社区发现任务的效率。

3.4 社区数量的确定

动态距离算法中会出现碎片化社区问题,导致算法的准确性降低并且无法准确估计社区数量。随机竞争学习中,网络粒子的数量对应社区的数量。在粒子达到收敛状态时,每个节点都会归属于相应粒子对应的社区,因此随机竞争学习中不会出现碎片化社区问题。随机竞争学习中使用网络粒子平均最大控制值指标来确定网络中合理的粒子数量,以此来获得社区的数量,网络粒子平均控制指标的定义如下:

$$\langle R(t) \rangle = \frac{1}{V} \sum_{u=1}^V \max_{m \in k} (\bar{N}_u^{(m)}(t)) \quad (23)$$

其中, $\langle R(t) \rangle$ 表示网络中粒子在 t 时刻对于节点的平均控制指标, $\bar{N}_u^{(m)}(t)$ 表示粒子 m 在 t 时刻对于节点 v_u 的控制能力。

用网络粒子平均最大控制指标来估计社区数量的基本思路是:对于一个拥有 k 个社区的网络,如果将 k 个粒子放入网络中,则每个粒子会控制一个社区。在粒子游走达到收敛状态时,粒子不会影响其他粒子所控制的区域,此时 $\langle R(t) \rangle$ 会取到最大值。而当粒子的数量大于 k 时,多个粒子共同影响一个社区的情况不可避免,因此网络粒子平均控制指标 $\langle R(t) \rangle$ 会减小;而当粒子的数量小于 k 时,粒子将会尝试控制多个社区,这样粒子对于节点的控制能力也会减小,相应的控制指标

$\langle R(t) \rangle$ 也减小。因此,为了找到网络中最佳的社区数量,随机竞争学习中网络中每次放置的粒子数量将从 2 逐渐增加至 $k+1$,然后记录不同粒子数量下粒子达到收敛状态时的 $\langle R(t) \rangle$ 值。当 $\langle R(t) \rangle$ 出现最大值时,此时网络中粒子的数量 k 即为最佳社区的数量。

3.5 DDSTL 算法框架

基于动态距离和随机竞争学习,本文提出了一种新的社区发现算法 DDSCL。其核心思想是先计算节点度值和节点间的欧氏距离,以确定粒子的初始位置,紧接着根据动态距离算法计算节点间的动态距离,粒子会结合节点间动态距离的游走概率在网络中进行游走。当粒子在网络中游走一段时间后,结合节点粒子控制力向量交互模式更新动态距离,粒子则依据更新后的动态距离继续游走。当网络中节点粒子控制力变化趋向于一个稳定值后,即粒子达到收敛状态,粒子结束游走,网络中的节点将被对其具有最大控制力的粒子占据。最后,网络中每一个粒子对应一个社区,根据各粒子占据的节点来揭示网络的社区结构。DDSCL 算法的大致过程如下:

- (1) 初始化粒子数量为 2;
- (2) 根据式(16)确定粒子的初始位置;
- (3) 根据式(14)计算节点间的初始动态距离;
- (4) 根据式(7)和式(22)开始粒子的游走;

(5) 当粒子游走一段时间后,结合节点粒子控制力向量交互模式,根据式(21)重新计算节点的动态距离,粒子根据更新后的动态距离继续在网络中进行游走;

(6) 当网络中节点粒子控制力向量趋向于一个稳定值后,即 $\|\bar{N}(t) - \bar{N}(t-1)\|_{\infty} < \epsilon$ 时,粒子停止游走;

(7) 根据式(23)计算 $\langle R(t) \rangle$ 值,判断 $\langle R(t) \rangle$ 值是否出现最大值,若是,则执行步骤(8);否则粒子数量加 1,继续执行步骤(2)~步骤(7);

(8) 网络中每一个粒子对应一个社区,此时网络中的粒子的数量即为实际社区数量。节点被对其具有最大控制力的粒子占据,根据每个粒子占据的节点来揭示社区结构。

DDSCL 算法的详细描述如算法 1 所示。

算法 1 DDSCL 算法

Input: 复杂网络 $G=(V, E)$, 游走概率 λ , 粒子能量变化 Δ , 粒子稳定值 ϵ , 独有邻居参数 ϕ , 节点粒子控制力向量影响参数 δ

Output: 社区数量 K , 网络社区发现结果

1. For 节点 $v_i \in V$ do
2. 根据式(16)计算粒子的初始位置 V_{ini}
3. End for
4. For 边 $e = \{u, v\} \in E$ do
5. 根据式(10)计算距离 DNI_e^0
6. For 节点 $z \in CN(e)$ do
7. 根据式(11)计算距离 CNI_z^0
8. End for
9. For 节点 $x \in EN(u)$ 节点 $y \in EN(v)$ do
10. 根据式(13)计算距离 ENI_e^0
11. End for
12. End for
13. 根据式(14)计算初始动态距离 Jac^0

```

14. 根据式(9)初始化随机竞争学习
15. Repeat
16.   K=2
17.   Repeat
18.     For k=1→K do//开始随机竞争学习
19.       根据式(7)和式(22)开始粒子的随机游走和优先游走,记录粒子位置 P(t)
20.     End for
21.     更新节点粒子控制力向量  $\bar{N}_i$ 、粒子能级 E(t)、粒子状态 S(t)
22.     t←t+1
23.     Do{
24.       根据式(21)将粒子控制力向量影响融入节点间动态距离计算,更新节点间的动态距离 Jaci
25.     }while(t=500N)
26.     until  $\bar{N}(t) - \bar{N}(t-1)_{\infty} < \epsilon$ 
27.     记录⟨R(t)⟩
28.     K←K+1
29. until ⟨R(t)⟩出现最大值
30. Return K-1,  $\bar{N}(t)$ 

```

3.6 时间复杂度分析

DDSCl 算法中确定节点初始位置需要计算节点度值和节点间 Jaccard 距离,因此时间代价为 $O(|E|)$;确定节点间初始动态距离时需要计算两个节点独有邻居节点对于节点间动态距离的影响,因此时间代价为 $O(d|E|)$,其中 d 为两个节点独有邻居节点的平均数量;在粒子游走过程中,需要计算每个粒子向邻居节点游走的概率,时间代价为 $O(\langle k \rangle K)$,其中 $\langle k \rangle$ 为网络节点平均度, K 为粒子的数量;计算节点粒子控制力向量的时间代价为 $O(K^2)$;网络中的节点至少被粒子访问一次,因此粒子游走总时间代价为 $O(\langle k \rangle KV + K^2V)$,由于复杂网络通常是稀疏的,因此 $\langle k \rangle \ll V$;动态距离更新过程中只需要使用上一步计算的动态距离,时间代价为 $O(T|E|)$, T 为动态距离更新次数;因此 DDSCl 算法的时间复杂度为 $O(|E| + d|E| + T|E| + \langle k \rangle KV + K^2V) \approx O(C(|V| + |E|))$,其中 C 为常数。动态距离是作为粒子优先游走概率的一个计算指标,因此动态距离的更新只需要根据网络的规模在粒子游走过程中进行少量更新, T 的取值通常不超过 5。DDSCl 算法的时间复杂度与网络中节点和边的数量呈线性关系,因此在运用到大规模网络运算时具有很好的扩展性。

4 实验

为测试 DDSCl 算法的可行性和有效性,本文首先分析算法的参数取值,然后将提出的算法在 8 个真实网络数据集上与常用算法进行对比。其中前 4 个数据集具有真实的网络社区发现情况,后 4 个数据集没有真实的网络社区发现情况。本文选取的对比算法分别是 CNM^[26], LPA^[27], Walktrap^[28], SCL^[7], Attractor^[8] 这 5 种算法。

CNM 是一种基于模块度函数合并的社区发现方法,通过不断合并两个社团来获得最大模块度函数值增益以实现社区发现任务,在基于模块度优化的社区发现算法中具有代表性。

LPA 算法是一种基于标签传播的社区发现算法,节点标签将会按照节点相似度传播给邻居节点,节点也会根据邻居节点的标签来更新自身标签,根据标签传播迭代结束后每个节点的标签来实现社区发现任务,LPA 算法由于其线性的时间复杂度和较好的社区发现效果,在近几年的基于网络动力学的社区发现算法中具有代表性。Walktrap 是一种基于随机游走的社区发现算法,Walktrap 认为节点在随机游走过程中会陷于社区内部,因此基于节点间的距离定义两个节点的相似性来实现社区划分,Walktrap 算法在速度和精度上都有出色的表现,是基于随机游走的社区发现算法中的代表性算法。SCL 是随机竞争学习算法,通过粒子竞争机制以无监督学习的方式来实现网络的社区发现任务,是当前基于网络动力学的社区发现算法中比较典型的算法。Attractor 是动态距离算法,以网络中节点收敛状态下的距离揭示网络社区结构,实验中可以取得不错的效果,已被广泛应用于社区发现任务中。

4.1 实验数据集介绍

实验中所使用的数据集的具体信息如表 1 所列。

表 1 真实网络数据集的基本信息

Datasets	Nodes	Edges	Community
Karate	34	78	2
Football	115	613	12
Dolphins	62	159	2
Polbooks	105	441	3
Jazz	198	5484	—
Email	1133	10902	—
Facebook	4039	88234	—
Dblp	317080	1049866	—

Karate 数据集来源于美国大学生空手道俱乐部网络^[29], Football 数据集来源于美国橄榄球足球联盟比赛网络^[30], Dolphins 数据集来源于海豚关系网络^[31], Polbooks 数据集来源于美国政治书籍网络^[32], Jazz 数据集来源于爵士乐舞者关系网络^[33], Email 数据集来源于电子邮件通信网络^[34], Facebook 数据集来源于 Facebook 平台上好友关系网络^[35], Dblp 数据集来源于科学家合作关系网络^[36]。

4.2 实验指标和实验环境介绍

本文使用的社区发现算法的评价指标为标准化互信息 NMI ^[37]和模块度 Q 值^[26],使用不同的指标对相应的数据集进行评估。

设真实社区发现结果定义为 A ,通过算法计算得到的结果定义为 B ,由此得到混淆矩阵 C , C 的行代表真实社区发现结果, C 的列代表算法计算的社区发现结果。 C_{ij} 代表在 A 的结果中为社区 i 的节点却在 B 中划分到社区 j 的节点的数量,则 NMI 的计算公式如下:

$$NMI(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{ij} \log_2 \left(\frac{C_{ij} N}{C_i C_j} \right)}{\sum_{i=1}^{C_A} C_i \log_2 \left(\frac{C_i}{N} \right) + \sum_{j=1}^{C_B} C_j \log_2 \left(\frac{C_j}{N} \right)} \quad (24)$$

其中, C_A 为网络真实社区发现结果的社区数量, C_B 为算法得到的社区发现结果的社区数量, N 为网络节点总数。 $NMI(A, B)$ 的取值为 $[0, 1]$,一般认为 NMI 的值越高,算法的社区发现效果就越好。

模块度 Q 是由 Clauset 等提出的社区发现评价指标,常用作算法在无真实社区发现情况数据集上的评价指标。模块度 Q 的定义如下:

$$Q = \frac{1}{2M} \sum_{i,j} \left[A_{ij} - \frac{d_i d_j}{2M} \right] \delta(c_i, c_j) \quad (25)$$

其中, M 是复杂网络中边的数量; A_{ij} 为网络的邻接矩阵; d_i 代表网络中节点的度数; c_i 代表节点 i 所属的社区。当节点 i 和节点 j 在不同社区时, $\delta(c_i, c_j)$ 为 0; 在同一社区时, $\delta(c_i, c_j)$ 为 1。模块度 Q 的取值范围一般是 $[0.3, 0.7]$, Q 值越大, 说明网络的社区发现效果越好。但是, 模块度函数存在“分辨率极限”的问题, 算法计算得到的社区规模越大, 模块度 Q 值往往越大, 容易出现超大型社区。

本文的实验环境为: 处理器为 Inter(R) Core(TM) i7-8750H CPU@2.20 GHz, 内存为 16 GB, 操作系统为 Windows 10 64 bit。

4.3 参数分析

随机竞争学习的参数取值参照文献[7], 游走概率参数 λ 取值为 0.6, 粒子能量变化参数 Δ 取值为 0.07, 粒子稳定值参数 ϵ 取值为 0.05。动态距离算法的独有邻居影响参数 ϕ 的取值参照文献[8], 取值为 $[0.4, 0.6]$ 。本文提出的 DDSCL 算法需要被讨论的参数是节点间粒子控制力向量对动态距离的影响程度 δ 。该参数可以决定粒子游走的过程对于动态距离的影响程度。当参数 δ 的取值非常低或非常高时, 无论粒子的游走情况如何, 节点间的动态距离都会不断减小或增加直至为 0 或 1, 这显然不符合动态距离的基本理论。因此, 当参数 δ 介于 $[0.2, 0.8]$ 之间时, 节点间粒子控制力向量对于动态距离的影响才会变得相对合理。

参数分析实验在 Karate, Football, Dolphins, Polbooks 这 4 个数据集上进行仿真验证, 参数 δ 介于 $[0.2, 0.8]$ 之间, 每个参数运行 5 次, 记录仿真结果的 NMI 值如图 2 所示。

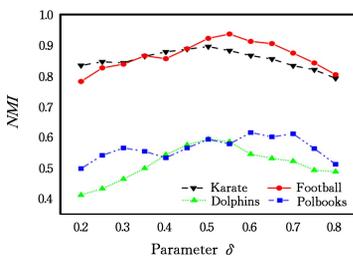


图 2 参数 δ 对于算法的影响

Fig. 2 Influence of parameter δ on the algorithm

由图 2 可知, 当 δ 为 $[0.5, 0.6]$ 之间时, NMI 值最高, 实验效果最好; 当 δ 趋向于 0 或者 1 时, NMI 值会变小。因此, 节点间粒子控制力的变化可以影响动态距离的变化, 当节点间粒子控制力向量影响程度取值合理时, 社区发现的准确性会提高。本文在接下来的实验中将参数 δ 设置为 0.5。

4.4 实验结果分析

(1) 有真实网络社区发现结果数据集上的测试结果和分析

将本文提出的 DDSCL 算法与对比算法在有真实网络社区发现结果数据集上进行测试, 并采用 NMI 评价指标作为社区发现评价标准。测试结果如表 2 和图 3 所示。

表 2 有真实网络社区发现结果数据集上算法的 NMI 值比较
Table 2 Comparison of NMI values of algorithms with real network community partition data set

Datasets	CNM	LPA	Walktrap	SCL	Attractor	DDSCL
Karate	0.705	0.699	0.566	0.821	0.859	0.894
Football	0.733	0.911	0.756	0.947	0.923	0.935
Dolphins	0.574	0.521	0.556	0.514	0.559	0.594
Polbooks	0.473	0.534	0.492	0.566	0.585	0.615

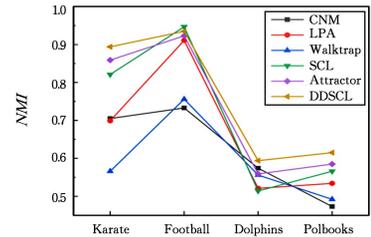


图 3 有真实网络社区发现结果数据集上算法的 NMI 值比较

Fig. 3 Comparison of NMI values of algorithms with real network community partition data set

从表 2 和图 3 可知, DDSCL 在 Karate, Dolphins, Polbooks 数据集上的 NMI 值最高, 并且相比其他算法有很大程度上的提升。在 Karate 数据集上相比 SCL 算法提升了 8.9%, 在 Dolphins 数据集上相比 Attractor 算法提升了 6.3%, 在 Football 数据集上的 NMI 值稍小于 SCL 算法。实验中, Attractor 算法的 NMI 值明显高于 SCL 算法, 通过分析其原因是数据集均属于小规模网络, 网络节点之间的连接紧密度高, Attractor 不容易出现小规模社区和孤立点, 而 SCL 算法中粒子游走的相对随机性使得社区发现的准确性相对偏低; 而结合动态距离的 DDSCL 算法有效地解决了这个问题。综上所述, DDSCL 算法划分的社区结果优于其他算法的划分结果, 降低了算法中粒子优先游走的随机性, 证明了算法的可行性和有效性。

(2) 无真实网络社区发现结果数据集上的测试结果和分析

将本文提出的 DDSCL 算法与对比算法在无真实网络社区发现结果数据集上进行测试, 社区发现评价指标为模块度 Q 值。测试结果如表 3 和图 4 所示。

表 3 无真实网络社区发现结果数据集上算法的模块度 Q 值比较
Table 3 Comparison of modularity Q values of algorithms with no real network community partition data set

Datasets	CNM	LPA	Walktrap	SCL	Attractor	DDSCL
Jazz	0.274	0.288	0.263	0.405	0.294	0.423
Email	0.499	0.468	0.423	0.513	0.275	0.515
Facebook	0.764	0.812	0.798	0.825	0.621	0.841
Dblp	0.721	0.633	0.594	0.741	0.213	0.784

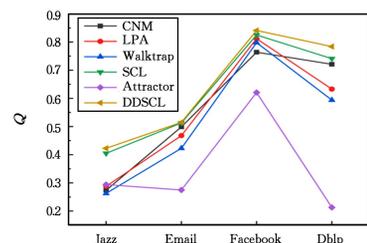


图 4 无真实网络社区发现结果数据集上算法的模块度 Q 值比较

Fig. 4 Comparison of modularity Q values of algorithms with no real network community partition data set

从表 3 和图 4 可知, DDSCL 算法在各个数据集上均取得了最高的模块度 Q 值。在 Jazz, Facebook, Dblp 数据集上, DDSCL 算法相比 SCL 算法分别提高了 4.5%, 1.9%, 5.8%, 可见 DDSCL 算法通过引入动态距离算法有效降低了 SCL 算法中粒子优先游走的随机性, 提高了社区发现效果。在 Facebook 数据集上, DDSCL 算法相比 Attractor 算法提升了 31.2%。Attractor 算法在各个数据集上的表现明显差于 SCL 算法和 DDSCL 算法。通过分析是由于数据集网络规模大且较为稀疏, 节点间的联系并不紧密, 导致 Attractor 算法出现大量碎片化社区, 因此模块度 Q 值偏低。Attractor 算法在去除碎片化社区后, 得到的模块度 Q 值会有大幅度的提升。综上所述, DDSCL 算法在无真实社区发现结果的数据集上的社区发现结果优于其他算法, 并且解决了动态距离算法出现的碎片化社区问题, 提高了社区发现的准确性, 综合表现性能最佳。

结束语 本文提出了一种融合动态距离和随机竞争学习的社区发现算法 DDSCL。该算法结合节点度值和节点间欧氏距离来确定随机竞争学习中粒子的初始位置, 使得不同粒子在游走初期不会在同一社区内进行竞争, 加快了粒子的收敛速度。同时, 将动态距离引入到随机竞争学习中, 使得随机竞争学习的粒子优先游走过程更加具有方向性, 将粒子游走的过程用于优化动态距离, 使得动态距离的计算更加准确。当粒子达到收敛状态时, 网络中的节点将被对其具有最大控制力的粒子占据。最后, 每一个粒子对应一个社区, 根据各粒子占据的节点来揭示网络的社区结构。通过在真实网络数据集上的实验结果表明, DDSCL 算法不仅降低了随机竞争学习中粒子优先游走的随机性, 而且解决了动态距离算法出现的碎片化社区问题, 提高了社区发现结果的准确性。与其他算法相比, DDSCL 算法是有效且可行的。由于本文的研究对象是静态非重叠网络, 没有考虑动态社区网络和重叠社区网络的划分问题, 在未来研究中将致力于对动态网络和重叠网络的研究。

参 考 文 献

- [1] FORTUNATO S. Community detection in graphs[J]. *Physics reports*, 2010, 486(3/4/5): 75-174.
- [2] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. *Proceedings of the National Academy of Sciences*, 2002, 99(12): 7821-7826.
- [3] RADICCHI F, CASTELLANO C, CECCONI F, et al. Defining and identifying communities in networks[J]. *Proceedings of the national academy of sciences*, 2004, 101(9): 2658-2663.
- [4] WANG Z, WANG C, LI X, et al. Evolutionary Markov Dynamics for Network Community Detection[J/OL]. *IEEE Transactions on Knowledge and Data Engineering*, 2020. <https://ieeexplore.ieee.org/abstract/document/9099469>.
- [5] BIAN Y, LUO D, YAN Y, et al. Memory-based random walk for multi-query local community detection [J]. *Knowledge and Information Systems*, 2020, 62(5): 2067-2101.
- [6] EL KOUNI I B, KAROUI W, ROMDLHANE L B. Node Importance based Label Propagation Algorithm for overlapping community detection in networks[J]. *Expert Systems with Applications*, 2020, 162: 113020.
- [7] SILVA T C, ZHAO L. Stochastic competitive learning in complex networks[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2012, 23(3): 385-398.
- [8] SHAO J, HAN Z, YANG Q, et al. Community detection based on distance dynamics[C]// *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2015: 1075-1084.
- [9] KOHONEN T. The self-organizing map[J]. *Proceedings of the IEEE*, 1990, 78(9): 1464-1480.
- [10] KOSKO B. Stochastic competitive learning[C]// *1990 IJCNN International Joint Conference on Neural Networks*. IEEE, 1990: 215-226.
- [11] CARPENTER G A, GROSSBERG S. ART 2: Self-organization of stable category recognition codes for analog input patterns [J]. *Applied Optics*, 1987, 26(23): 4919-4930.
- [12] QUILES M G, ZHAO L, ALONSO R L, et al. Particle competition for complex network community detection[J]. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2008, 18(3): 033107.
- [13] SILVA T C, ZHAO L. Detecting and preventing error propagation via competitive learning[J]. *Neural Networks*, 2013, 41: 70-84.
- [14] VERRI F A N, URIO P R, ZHAO L. Network unfolding map by vertex-edge dynamics modeling[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2016, 29(2): 405-418.
- [15] RODRIGUES R D, ZHAO L, ZHENG Q, et al. Structural outlier detection: A tourist walk approach[C]// *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, 2017: 382-387.
- [16] GAO X, ZHENG Q, VERRI F A N, et al. Particle competition for multilayer network community detection[C]// *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*. 2019: 75-80.
- [17] LI W, GU Y, YIN D, et al. Research on the community number evolution model of public opinion based on stochastic competitive learning[J]. *IEEE Access*, 2020, 8: 46267-46277.
- [18] PASSERINI J A R, BREVE F. Complex Network Construction for Interactive Image Segmentation Using Particle Competition and Cooperation: A New Approach[C]// *International Conference on Computational Science and Its Applications*. Cham: Springer, 2020: 935-950.
- [19] LI J H, WANG C D, LI P Z, et al. Discriminative metric learning for multi-view graph partitioning[J]. *Pattern Recognition*, 2018, 75: 199-213.
- [20] SUN H, CH'NG E, YONG X, et al. A fast community detection method in bipartite networks by distance dynamics[J]. *Physica A: Statistical Mechanics and its Applications*, 2018, 496: 108-120.
- [21] SUN H, JIA X, HUANG R, et al. Distance dynamics based overlapping semantic community detection for node-attributed networks[J/OL]. *Computational Intelligence*, 2020. <https://onlinelibrary.wiley.com/doi/abs/10.1111/coin.12324>.

- [22] WAN J, HAN D, YANG Z, et al. An improved algorithm for detecting community defined by node-to-node dynamic distance [J]. *International Journal of Modern Physics C (IJMPC)*, 2020, 31(11):1-18.
- [23] HENNIG C, HAUSDORF B. Design of dissimilarity measures: A new dissimilarity between species distribution areas[M]// *Data Science and Classification*. Berlin, Heidelberg: Springer, 2006: 29-37.
- [24] DANIELSSON P E. Euclidean distance mapping[J]. *Computer Graphics and Image Processing*, 1980, 14(3):227-248.
- [25] REAL R, VARGAS J M. The probabilistic basis of Jaccard's index of similarity[J]. *Systematic Biology*, 1996, 45(3):380-385.
- [26] CLAUSET A, NEWMAN M E J, MOORE C. Finding community structure in very large networks[J]. *Physical Review E*, 2004, 70(6):066111.
- [27] RAGHAVAN U N, ALBERT R, KUMARA S. Near linear time algorithm to detect community structures in large-scale networks[J]. *Physical Review E*, 2007, 76(3):036106.
- [28] PONS P, LATAPY M. Computing communities in large networks using random walks[C]// *International Symposium on Computer and Information Sciences*. Berlin, Heidelberg: Springer, 2005:284-293.
- [29] ZACHARY W W. An information flow model for conflict and fission in small groups[J]. *Journal of Anthropological Research*, 1977, 33(4):452-473.
- [30] GIRBAN M, NEWMAN M E J. Community structure in social and biological networks[J]. *Proceedings of the National Academy of Sciences*, 2002, 99(12):7821-7826.
- [31] LUSSEAU D, SCHNEIDER K, BOISSEAU O J, et al. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations [J]. *Behavioral Ecology and Sociobiology*, 2003, 54(4):396-405.
- [32] ADAMIC L A, GLANCE N. The political blogosphere and the 2004 US election:divided they blog[C]// *Proceedings of the 3rd International Workshop on Link Discovery*. 2005:36-43.
- [33] GUIMERA R, DANON L, DIAZ-GUILERA A, et al. Self-similar community structure in a network of human interactions[J]. *Physical Review E*, 2003, 68(6):065103.
- [34] LESKOVEC J, KLEINBERG J, FALOUTSOS C. Graph evolution: Densification and shrinking diameters[J]. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 2007, 1(1):2.
- [35] MCAULEY J J, LESKOVEC J. Learning to discover social circles in ego networks[C]// *NIPS*. 2012:539-547.
- [36] YANG J, LESKOVEC J. Defining and evaluating network communities based on ground-truth[J]. *Knowledge and Information Systems*, 2015, 42(1):181-213.
- [37] WHITLEY D, STARKWEATHER T, BOGART C. Genetic algorithms and neural networks: Optimizing connections and connectivity[J]. *Parallel Computing*, 1990, 14(3):347-361.



WANG Ben-yu, born in 1998, postgraduate. His main research interests include complex network and deep learning.



GU Yi-jun, born in 1968, Ph.D, professor, Ph.D supervisor. His main research interests include complex network and deep learning.