

基于社区密度的重叠社区检测算法

赵洋洋*, 刘士虎#, 胡志涛

云南民族大学, 数学与计算机科学学院, 云南 昆明

收稿日期: 2023年5月28日; 录用日期: 2023年6月23日; 发布日期: 2023年6月30日

摘要

社区检测是研究复杂网络的热点话题之一, 其目的是找到具有共同性质的社区成员。随着网络复杂性的增强, 传统算法在时间和精度上都受到了一定的限制。基于此, 本文提出一种基于社区密度的重叠社区检测方法。在该方法中, 我们首先对顶点的邻居集做交集运算, 在确定初始候选社区的基础上过滤掉社区密度低于阈值的社区。其次, 以社区间相似性这一概念作为社区合并的判断条件, 生成重叠社区。最后, 我们在两类合成网络上与现有的经典算法作比较, 实验结果表明本文所建立的算法是有效的。

关键词

社区检测, 重叠社区, 社区密度, 单一阈值

Overlapping Community Detection Algorithm Based on Community Density

Yangyang Zhao*, Shihu Liu#, Zhitao Hu

School of Mathematics & Computer Science, Yunnan Minzu University, Kunming Yunnan

Received: May 28th, 2023; accepted: Jun. 23rd, 2023; published: Jun. 30th, 2023

Abstract

Community detection is one of the hot topics in the study of complex networks, the purpose is to find community members with common properties. With the increasing complexity of networks, traditional algorithms are limited in both time and accuracy. In this paper, we propose a community density-based overlapping community detection method. In this method, we first intersect the set of neighbors of the vertices and filter out the communities that are below the community density threshold to determine the initial candidate communities. Next, the concept of inter-community

*第一作者。

#通讯作者

similarity was used as a criterion for community merging to generate overlapping communities. We finally compare the algorithm with existing classical algorithms on two types of synthetic networks and the experimental results show that the algorithm established in this paper is efficient.

Keywords

Community Detection, Overlapping Community, Community Density, Single Threshold

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

现实世界中的很多系统都可以抽象为复杂网络[1]，如蛋白质交互网络[2]、交通网络[3]、科学合著网络[4]等。在此类网络中，一般用顶点表示抽象实体，用边表示抽象实体之间的关联。随着复杂网络系统逐渐呈现多样化，挖掘和利用网络信息变的越来越困难。2004年 Newman 和 Girvan 首次提出复杂网络具有社区结构的特征[5]，为发掘复杂网络数据提供了新的思路。通过利用网络中顶点和边的结构特征，挖掘能够反映现实的网络信息，许多社区发现算法陆续被提出[6]。但大多数方法默认每个顶点只能隶属于唯一社区，事实上，真实网络中的顶点会存在多重隶属关系。例如，一个人会与多个社会群体存在关联，如家庭，朋友，同事等，这类问题被称作重叠社区检测问题[7]。

复杂网络数据挖掘工作目前遇到的主要挑战之一，就是识别更接近现实世界的重叠社区。截至目前，已经被提出来的用于重叠社区检测的算法大致可以分为五类。第一类方法是基于团渗透理论的算法，如 CPM [8]算法，谱分析聚类算[9]，非负矩阵分解法[10]等。此类方法类似于模式识别的方法，旨在发现网络特定的局部结构，但上述方法都需要预先给定社区数目，且对于大规模网络算法时间复杂度较高。第二类是基于链接划分的算法，Ahn 等人于 2010 年提出基于边相似性的层次聚类算法[11]，层次聚类算法通过计算顶点对间的相似性并创建带有层次的聚类树，然后对聚类树进行划分。虽然此类通过链接划分的方法看起来很直观，但基于链接的划分的算法将社区定义为模糊的，所以在划分质量上并不能得到合理的论证。第三类方法是基于模糊检测的算法[12]，不难发现重叠社区的定义与模糊集的定义十分相似，即每个元素对多个集合都有一定的隶属度。基于模糊思想的算法问题在于需要确定隶属向量的维数，将该值作为参数传递给算法，因此在计算过程中较为复杂。第四类方法为基于局部扩展的算法[13]，首先随机选择初始的种子顶点，然后构建适应度函数，将最优适应度函数的顶点添加到社区中，反之将其移除，反复进行此过程，直至不能进一步优化划分结果。第五类方法为标签传播算法，如 COPRA [14]、SLPA [15] 等，此类算法通过更新邻居顶点标签来改变社区的标签，此类算法减少了运行时间但形成的社区结果不稳定。

以上描述的五类算法适用于不同的网络，但随着互联网的飞速发展，网络的规模也在飞速扩张，顶点间的关系也表现的越来越复杂，这使得算法运行时间的增加，划分的结果变差。因此，本文提出一种基于社区密度的重叠社区检测算法(Community Density Based Overlapping Community Detection Algorithm, CDOCD)。首先通过社区密度函数和阈值确定候选社区，然后引入社区间相似性的概念作为合并候选社区和现有社区的判断条件。本算法不仅考虑了网络的结构特征，且在判断过程用使用单一阈值，在减少运行时间的前提下提高了划分精度。在两类人工合成网络上进行了充分的测试，并与其它重叠社区检测

算法进行了比较，在扩展的标准化互信息和时间复杂度上都呈现出良好的算法性能。

2. 预备知识

从数学的角度，复杂网络可以表示为一个图 $G=(V,E)$ ，其中 $V=\{v_1,v_2,\dots,v_n\}$ 表示顶点的集合， $E=\{e_1,e_2,\dots,e_m\}$ 表示所有边的集合。其中 $n=|V|$ 和 $m=|E|$ 为图中顶点数和连边数。进一步，用 $N(v)=\{v\}\cup\{u|(u,v)\in E\}$ 表示任意 $v\in V$ 的邻居顶点集。

定义 1 (社区间相似性) 给定图 G 及任意两个社区 $C_1, C_2 \subseteq G$ ，定义社区 C_1, C_2 之间的相似性为

$$Sim(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}, \quad (2-1)$$

其中 $|C_1 \cap C_2|$ 表示 C_1, C_2 两个社区中相同元素的个数， $|C_1 \cup C_2|$ 表示两个社区中不重复出现的总元素个数。

定义 2 (社区内密度) 对于任意社区 C ，其社区内密度定义为

$$\rho_{in}(C) = \frac{k_{in}}{n_c(n_c - 1)/2}, \quad (2-2)$$

其中 k_{in} 表示社区 C 的内部度数和，即社区内部连边数， n_c 表示社区 C 的顶点数目， $\rho_{in}(C)$ 用于衡量社区内部联系紧密程度，取值范围在 $[0,1]$ 之间， $\rho_{in}(C)$ 越大表示社区 C 内部联系越紧密。

定义 3 (社区外密度) 对于任意社区 C ，其社区外密度定义为

$$\rho_{out}(C) = \frac{k_{out}}{n_c(n - n_c)}, \quad (2-3)$$

其中 k_{out} 表示社区 C 外部度数和，即社区内顶点与社区外顶点连边数， $\rho_{out}(C)$ 用于衡量社区外部联系紧密程度，取值范围在 $[0,1]$ 之间， $\rho_{out}(C)$ 越大表示社区 C 内部联系越稀疏。

定义 4 (社区密度) 对于任意社区 C ，社区密度定义为

$$\rho(C) = \rho_{in}(C) - \rho_{out}(C), \quad (2-4)$$

其中 $\rho(C)$ 用于评估社区总体紧密程度，取值范围在 $[0,1]$ 之间， $\rho(C)$ 越大表示社区质量越高。

社区实质上是由顶点和边组成的一个具有共同性质的顶点集合，集合内部顶点联系紧密，不同集合之间联系稀疏。因此在最大化 $\rho_{in}(C)$ 和最小化 $\rho_{out}(C)$ 寻找一个平衡，是社区发现算法的目标。在本文中我们 $\rho(C)$ 和阈值来判断生成社区的质量，在很大程度上提高了社区划分结果的质量。

3. 基于社区密度的重叠社区检测算法

本节主要介绍基于社区密度的重叠社区检测算法，具体工作分为如下部分：重叠社区检测算法的构建和对该算法时间复杂度的分析。算法构建部分，我们主要从邻域扩展和社区合并两个层面加以说明。

3.1. 算法构建

我们将 CDOCD 算法分为邻域扩展和社区合并两个主要阶段。首先在邻域扩展阶段我们将邻居顶点视为最佳候选顶点，通过邻居顶点间的交集运算确定候选社区，且候选社区密度要不小于给定阈值 t 以确保生成的候选社区的质量。然后进入社区合并阶段，在此阶段我们将生成的候选社区与现有社区进行合并。为确保合并后的社区是联系紧密且相似的，合并的两个社区其社区间相似性也要大于阈值 t ，并将合并后的社区作为新的候选社区继续进行迭代。在算法两个阶段中我们都采用的唯一阈值，这样可以避

免重复定义阈值增加算法的复杂度。

算法 1 为基于社区密度的重叠社区检测算法的第一阶段，即邻域扩展阶段。在此阶段通过邻居顶点间的交集运算确定候选社区，考虑了网络的结构性质，并利用社区密度和阈值确保生成社区的质量。此过程迭代进行，直至社区划分结果不满足更新后的社区密度和阈值。

算法 1 基于社区密度的重叠社区检测算法（邻域扩展阶段）

输入：图数据 $G=(V, E)$ ，阈值 t ，其中 $t \in [0, 1]$ 。

输出：重叠社区集合 $C = \{C_1, C_2 \dots C_m\}$ 。

过程：

- 1、计算顶点集 V 中所有顶点 v 的邻居顶点集 $N(v)$ ；
- 2、计算邻居顶点 v_n 的邻居顶点集 $N(v_n)$ ，其中 $v_n \in N(v)$ ；
- 3、将 $N(v)$ 和 $N(v_n)$ 的交集作为候选社区 C_c ；
- 4、根据公式(2-4)计算此时社区密度 $\rho(C_c)$ ，并要求 $\rho(C_c)$ 不小于当前阈值 t ；
- 5、更新此时的社区集合 C 、候选社区 C_c 、阈值 t 。

显然，在算法 1 中生成的候选社区并不是最终的社区划分结果，需要进行社区合并扩展候选社区生成最终的重叠社区集合。算法 2 为基于社区密度的重叠社区检测算法的第二阶段，即社区合并阶段。在此阶段我们讨论了候选社区 C_c 和现有社区 C_n 合并的几种情况，其中 $C_n \subseteq C$ 。

算法 2 基于社区密度的重叠社区检测算法（社区合并阶段）

输入：候选社区 C_c ，阈值 t ，其中 $t \in [0, 1]$ 。

输出：重叠社区集合 $C = \{C_1, C_2 \dots C_m\}$ 。

过程：

- 1、若候选社区 C_c 是现有社区 C_n 的子集或者等于现有社区，则返回社区列表 C ；
- 2、若现有社区 C_n 是候选社区 C_c 的真子集，则将 C_n 从社区列表 C 中移除；
- 3、候选社区 C_c 不是现有社区 C_n 的子集或超集，根据公式(2-1)计算候选社区 C_c 与现有社区 C_n 间的相似性，其中 $C_n \subseteq C$ ；
- 4、若候选社区与现有社区间相似性大于阈值，合并两个社区生成新的候选社区 C_c ；
- 5、根据公式(2-4)计算更新后候选社区密度，若这时候候选社区密度 $\rho(C_c)$ 不小于阈值；
- 6、将新的候选社区 C_c 添加至重叠社区集合 C 。

在社区合并阶段我们将候选社区与现有社区合并生成新的候选社区，然而两个相似度较低的社区合并会生成较大的且相关性的较低社区，或者会丢失部分小的社区，从而降低社区划分精度。因此新的候选社区的社区密度要不小于阈值 t ，并且为保证合并后的社区内部联系是紧密且相似的，合并的两个社区间相似性也要大于阈值 t 。相似性的阈值沿用社区密度阈值 t ，这种方法可以避免重复定义阈值，减少算法的复杂性。

3.2. 算法复杂度分析

在本节我们对 CDODC 算法的时间复杂度给出具体分析。给定一个顶点数为 n ，边数为 m 的图 G ，平均每个顶点有 $2m/n$ 条边，因此遍历顶点和邻居顶点耗时为 $O(m)$ 。算法通过集合运算生成社区，为线性时间，运行的平均时间为 $O(1)$ 。社区密度是通过遍历候选社区的顶点进行计算的，而候选社区是由两个顶点的邻居顶点的交集所形成的，所以候选社区的元素的平均个数为 $2m/n$ ，因此在社区密度函数上运行的时间为 $O(m^2/n^2)$ 。社区合并阶段需要遍历所有社区通过子集运算查找两个集合的相同元素，社区个数为 $n/(2m/n)$ ，因此社区合并阶段的运行时间为 $O(n^2/2m \times (2m/n + m^2/n^2)) = O(m/2 + n)$ 。综上，对所

有顶点及其邻居顶点重复以上时间, 需要遍历 $O(m)$ 次, 所以 $O(m^3/n^2)$ 为算法总运行时间。

4. 实验分析

在现实世界中, 许多数据的获取是有一定困难的, 或者获取的数据集真实划分是不确定的, 因此可以通过合成数据进行检验和评估算法的有效性。比如广被接受的重叠图生成器 OCG [16]和 LFR [17], 在本节我们通过人工生成数据对算法进行测试, 选择可用于重叠社区评估的归一化互信息检验算法精度。

4.1. 评估指标

用于比较算法性能的方法很多, 本文采用了扩展到重叠社区的归一化互信息[18], 数学上表示为

$$NMI(C, G) = \frac{-2 \sum_{i=1}^{|C|} \sum_{j=1}^{|G|} N_{ij} \log \left(\frac{N_{ij} N}{N_i N_j} \right)}{\sum_{i=1}^{|C|} N_i \log \left(\frac{N_i}{N} \right) + \sum_{j=1}^{|G|} N_j \log \left(\frac{N_j}{N} \right)}, \quad (4-1)$$

其中 N_i 表示 C_i 中的顶点数, N_{ij} 表示社区 C_i 中出现在真实社区 G_j 中的顶点数, $|C|$ 表示检测到的社区数, $|G|$ 表示实际社区数。NMI 取值范围为 $[0, 1]$, 其中 1 表示检测到社区与实际社区结构完全一致。

4.2. 算法性能评估

由第 3 节可知, 阈值的选择是确保算法成功的关键, 因此在与其它算法进行比较之前, 通过改变 OCG 网络和 LFR 网络的参数来评估阈值对算法精度的影响。

4.2.1. 改变人工合成网络参数评估阈值对算法精度的影响

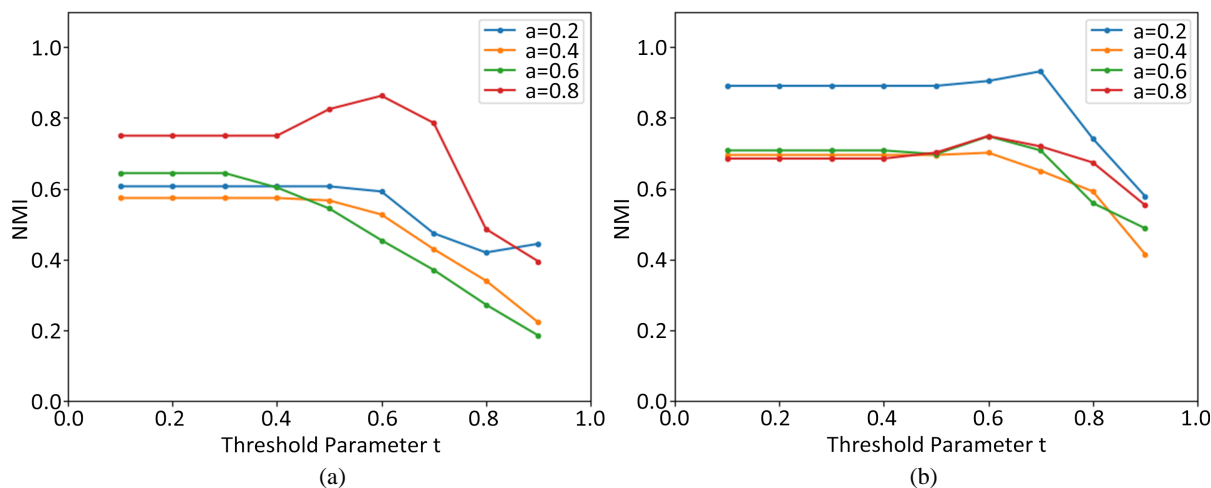


Figure 1. Variation of algorithm accuracy with threshold value for changing OCG network parameters

图 1. 改变 OCG 网络参数算法精度随阈值的变化

在图 1(a)中, 我们设置 OCG 网络的社区个数 i 为 10, 每个社区的平均顶点数 a 为 100, 重叠比率 c 为 0.3, 连接比率 λ 为 0.3, 固定社区外密度 ω 为 0.2, 调整社区内密度 α 从 0.2 至 0.8。我们发现算法精度随着社区内密度的增加而增加, 并且只要阈值参数的上界是社区内密度, 就能够得到较高的 NMI 评分。

图 1(b)的 OCG 网络其他参数不变, 固定社区内密度 α 为 0.8, 调整社区外密度 ω 从 0.2 至 0.8, 我们发现

算法性能随阈值 t 的增加而下降，并且随着 ω 的增加，算法精度下降，这是由于随着社区密度外密度的增加，社区密度减小，伴随着阈值的提高，候选社区无法满足此时的社区密度，许多顶点无法得到正确的划分，使得算法性能下降。

Table 1. LFR network parameters setting
表 1. LFR 网络参数设置

网络	参数
T_1	$N = 1000, k = 12, k_{\max} = 24, S_{\min} = 15, S_{\max} = 25, O_n = 300, O_m = 2, \mu = 0.15 - 0.45$
T_2	$N = 1000, k = 12, k_{\max} = 24, S_{\min} = 15, S_{\max} = 25, O_n = 200 - 500, O_m = 2, \mu = 0.3$
T_3	$N = 1000, k = 12, k_{\max} = 24, S_{\min} = 15, S_{\max} = 25, O_n = 300, O_m = 2 - 6, \mu = 0.3$

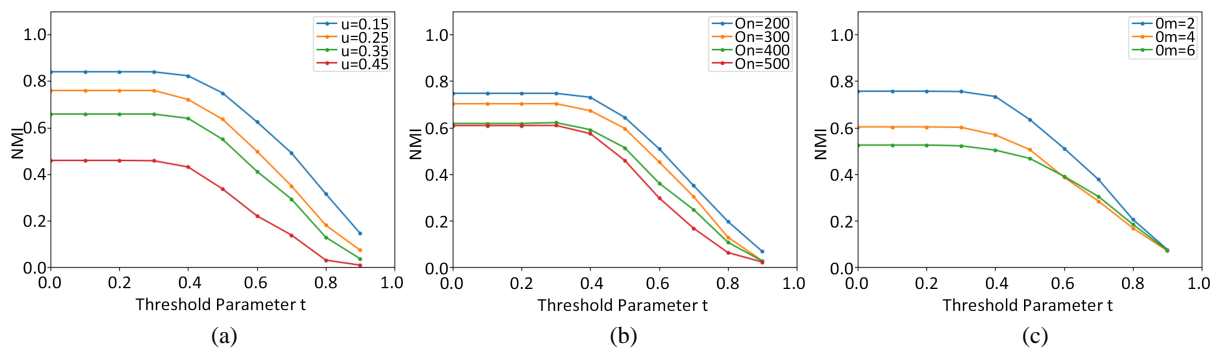


Figure 2. Variation of algorithm accuracy with threshold value for changing LFR network parameters
图 2. 改变 LFR 网络参数算法精度随阈值的变化

图 2(a)为表 1 中的 T_1 网络算法精度随阈值参数的变化。可以看出随着 μ 的增大，算法性能下降，但不明显。首先阈值在 0 到 0.4 时算法性能稳定，然后随着阈值的增加算法性能下降。这是因为社区合并阶段，我们依据阈值来判断两个社区间的相似性，而过高的阈值，会导致许多社区无法进行正常的合并，从而降低了算法精度。图 2(b)为 T_2 网络算法精度随阈值参数的变化。我们设置重叠顶点 O_n 由 200 至 500，步长为 100。随着重叠顶点数目增加算法性能下降，但相对稳定，说明该算法可以很好的适用于重叠社区的划分。图 2(c)为 T_3 网络算法精度随阈值参数的变化。我们设置重叠社区资格数 O_m 由 2 至 6，步长为 2。可以看出随着 O_m 的增加算法性能下降，这是因为随着顶点隶属社区数目增加，在社区合并阶段部分此类顶点无法得到正确的划分从而导致算法精度下降，但可依照经验根据数据类型改变阈值来提高划分精度。

通过上述实验可以看出，如果一个社区内部联系紧密，外部联系稀疏，通常选择较大的阈值，反之则选择较低的阈值。对于大型网络，我们可以通过提取随机社区获取他们的平均社区内外密度。对于小型网络，可以利用算法在一定阈值范围内进行运行选择合适的阈值。

4.2.2. 对比算法实验

为了进一步验证算法的有效性，我们将 CDOCD 算法和 COPRA、NISE [19]和 CPM 算法进行了比较。选择以上算法的原因如下：CORPA 算法是一种基于标签传播的重叠社区检测算法，在应用上和结果的准确性上与 CDOCD 算法相似，NISE 是通过贪婪方法拓展种子节点生成社区的方法，比较新的且有效的算法之一，CPM 算法，是一种通过派系来识别网络社区结构的算法，是重叠社区检测最常见的方法之一。

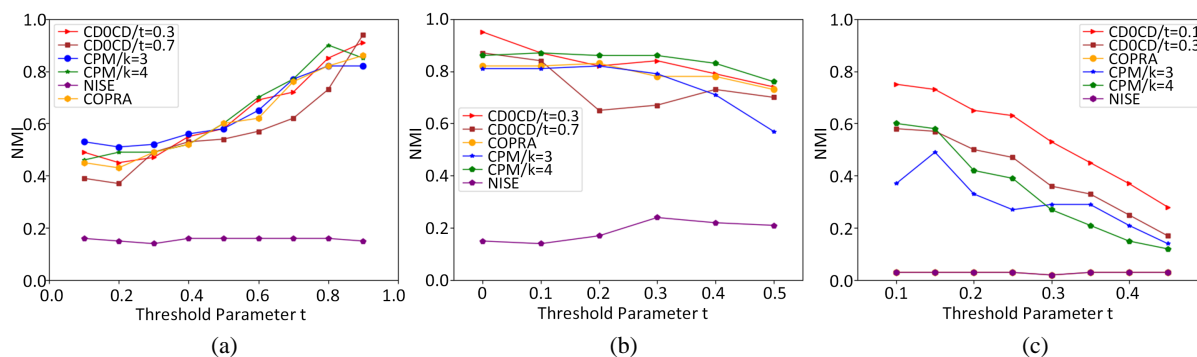


Figure 3. Comparison of algorithm accuracy under different networks

图 3. 不同网络下算法精度对比

在图 3(a)中, 我们设置 OCG 网络社区个数 i 为 10, 每个社区的平均顶点数 a 为 100, 重叠比率 ε 为 0.3, 连接比率 λ 为 0.3, 固定社区外密度 ω 为 0.2, 增加社区内密度 α 由 0.1 至 0.9, 步长为 0.1, 在该数据下每种算法运行 30 次取 NMI 均值。可以看出 CDOCD 算法同其他算法的性能都随着社区内密度的增加算法性能提高, 表明网络内部联系越强算法性能越佳。图 3(b)中其他参数保持不变, 固定社区内密度 α 为 0.8, 增加社区外密度 ω 由 0 至 0.5, 步长为 0.1。在该数据下每种算法运行 30 次取 NMI 均值, 通过此实验同样验证了, 通过优化社区密度能够提高社区划分精度。图 3(c)我们设置 LFR 网络顶点数 $N = 1000$, 重叠顶点数 $O_n = 500$, 重叠社区资格数 $O_m = 2$, 增加混合参数 μ 由 0.1 至 0.45, 步长为 0.05。在该数据下每种算法运行 10 次取 NMI 的均值, 可以看出每种算法随着 μ 值增加算法性能下降, 但 CDOCD 算法性能优于其他算法。

4.3. 算法时间复杂度对比

在本节主要对比 CDOCD 算法与 CPM 算法、COPRA 算法、NISE 算法的时间复杂度。

如表 2 所示, CDOCD 算法时间复杂度为 $O(m^3/n^2)$, 为多项式时间, 在大多数图中网络的边数大于顶点的数目, 因此 CDOCD 算法, 通过边来主导运行时间; CPM 算法的时间复杂度主要取决于网络的规模和社区结构两个因素。假设网络中有个节点, m 条边, k 个社区, CPM 算法的时间复杂度为 $O(m^k)$ 。在实际应用中, CPM 算法需要通过不断的搜索、扩展和组合来获取所有大小为 k 的团, 并且 k 的取值与网络的规模有关, 当 k 较大时, 算法的运行时间也会相应地增加。综合来看 CPM 算法时间复杂度较高, 适用于网络规模较小而且社区结构比较简单的情况; COPRA 算法通过给每个顶点设置标签列表解决 LPA [20]算法随机不稳定的问题。算法的时间复杂度主要取决于两个因素, 标签传播迭代的次数和每个节点的邻居节点数。具体时间复杂度为 $O(T * D * N)$, 其中 T 是迭代次数, D 是平均节点度数, N 是节点数量。在实际的实验过程中, 通常需要几十次迭代就能让大部分节点归属于一个社区, 并且具体运行时间还取决于具体数据集的大小和稠密程度, 综合来看该算法具有较快的运行速度, 然而算法会随着网络规模的增加迭代时间加长, 结果随机性也会随之增强; NISE 算法主要包括种子扩展和邻居节点加权两个步骤, 其时间复杂度主要与网络规模和算法参数有关。种子扩展阶段耗时 $O(k * L_1 * n)$, 其中 k 为种子数量, L_1 为每个种子扩展的最大层数, n 为网络节点数。对邻居节点加权耗时 $O(k * L_2 * n)$, 其中 L_2 为每个社区邻居节点选择的最大数量。该步骤的时间复杂度与种子扩展的时间复杂度相近因此 NISE 算法的总时间复杂度为 $O(k * L * n)$, 其中 L 为扩展的最大层数, k 为种子数量。NISE 算法在大型网络上运行稳定性表现良好, 但其运行步骤较多更加耗时, 综合来看 CDOCD 算法在运行时间上表现良好。

Table 2. Algorithm time complexity comparison
表 2. 算法时间复杂度对比

算法	时间复杂度
CDOCD	$O(m^3/n^2)$
CPM [8]	$O(m^k)$
COPRA [14]	$O(T * D * N)$
NISE [19]	$O(k * L * n)$

5. 结论

本文提出一种基于社区密度的重叠社区检测算法 CDOCD，一方面，通过设置单一阈值 t 作为算法的判断条件，减少算法的运行时间；另外我们将算法分为两个阶段，在每个阶段都将社区密度作为候选社区的生成条件确保社区划分的质量，提高了划分精度。并在两类人工合成网络上进行严格的实验，实验结果表明本文提出的算法在计算效率和社区划分质量上优于其他算法。在未来，我们将该算法应用于各类复杂网络真实数据以及大型网络数据上，更加高效且高质量地获取网络的信息。

基金项目

国家自然科学基金资助项目(61966039)。

参考文献

- [1] 安沈昊, 于荣欢. 复杂网络理论研究综述[J]. 计算机系统应用, 2020, 29(9): 26-31.
- [2] Chen, B.L., Fan, W.W., Liu, J. and Wu, F.-X. (2014) Identifying Protein Complexes and Functional Modules—From Static PPI Networks to Dynamic PPI Networks. *Briefings in Bioinformatics*, **15**, 177-194. <https://doi.org/10.1093/bib/bbt039>
- [3] Kolpin, D., Ramos, F.M.V., Veríssimo, P.E., *et al.* (2015) Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, **103**, 14-76. <https://doi.org/10.1109/JPROC.2014.2371999>
- [4] 张继洋, 李宁. 科学合著网络研究进展分析[J]. 情报理论与实践, 2012, 35(4): 124-128.
- [5] Clauset, A., Newman, M.E.J. and Moore, C. (2004) Finding Community Structure in Very Large Networks. *Physical Review E*, **70**, Article ID: 066111. <https://doi.org/10.1103/PhysRevE.70.066111>
- [6] Fortunato, S. (2010) Community Detection in Graphs. *Physics Reports*, **486**, 75-174. <https://doi.org/10.1016/j.physrep.2009.11.002>
- [7] Ye, F.H., Chen, C., Zheng, Z.B., Li, R.-H. and Yu, J.X. (2019) Discrete Overlapping Community Detection with Pseudo Supervision. 2019 *IEEE International Conference on Data Mining (ICDM)*, Beijing, 8-11 November 2019, 708-717. <https://doi.org/10.1109/ICDM.2019.00081>
- [8] Palla, G., Derényi, I., Farkas, I. and Vicsek, T. (2005) Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society. *Nature*, **435**, 814-818. <https://doi.org/10.1038/nature03607>
- [9] Zhang, Y., Levina, E. and Zhu, J. (2020) Detecting Overlapping Communities in Networks Using Spectral Methods. *SIAM Journal on Mathematics of Data Science*, **2**, 265-283. <https://doi.org/10.1137/19M1272238>
- [10] 李向利, 范学珍, 逯喜燕. 基于非负矩阵分解的修正模糊聚类算法[J]. 吉林大学学报(理学版), 2022, 60(6): 1416-1422.
- [11] Ahn, Y.-Y., Bagrow, J. and Lehmann, S. (2010) Link Communities Reveal Multiscale Complexity in Networks. *Nature*, **466**, 761-764. <https://doi.org/10.1038/nature09182>
- [12] Feng, O.G., Shen, Y., Xu, H.L., Jiang, H.Y. and Ren, S.G. (2021) Local Fuzzy Community Detection in Networks. *Recent Advances in Electrical & Electronic Engineering*, **14**, 122-129. <https://doi.org/10.2174/2352096513999200818104926>
- [13] Kun, G., Huang, X.T., Wu, L. and Chen, Y.Z. (2022) Local Community Detection Algorithm Based on Local Modularity Density. *Applied Intelligence*, **52**, 1238-1253. <https://doi.org/10.1007/s10489-020-02052-0>

-
- [14] Gregory, S. (2010) Finding Overlapping Communities in Networks by Label Propagation. *New Journal of Physics*, **12**, Article ID: 103018. <https://doi.org/10.1088/1367-2630/12/10/103018>
- [15] Xie, J. and Szymanski, B.K. (2012) Towards Linear Time Overlapping Community Detection in Social Networks. In: Tan, P.-N., Chawla, S., Ho, C.K. and Bailey, J., Eds., *Advances in Knowledge Discovery and Data Mining. PAKDD 2012. Lecture Notes in Computer Science*, Vol. 7302, Springer, Berlin, 25-36. https://doi.org/10.1007/978-3-642-30220-6_3
- [16] Becker, E., Robisson, B., Chapple, C.E., Guénoche, A. and Brun, C. (2012) Multifunctional Proteins Revealed by Overlapping Clustering in Protein Interaction Network. *Bioinformatics*, **28**, 84-90. <https://doi.org/10.1093/bioinformatics/btr621>
- [17] Lancichinetti, A. and Fortunato, S. (2009) Benchmarks for Testing Community Detection Algorithms on Directed and Weighted Graphs with Overlapping Communities. *Physical Review E*, **80**, Article ID: 016118. <https://doi.org/10.1103/PhysRevE.80.016118>
- [18] Shen, H.W., Cheng, X.Q., Cai, K. and Hu, M.-B. (2009) Detect Overlapping and Hierarchical Community Structure in Networks. *Physica A: Statistical Mechanics and Its Applications*, **388**, 1706-1712. <https://doi.org/10.1016/j.physa.2008.12.021>
- [19] Whang, J.J., Gleich, D.F. and Dhillon, I.S. (2016) Overlapping Community Detection Using Neighborhood-Inflated Seed Expansion. *IEEE Transactions on Knowledge and Data Engineering*, **28**, 1272-1284. <https://doi.org/10.1109/TKDE.2016.2518687>
- [20] Raghavan, U.N., Albert, R. and Kumara, S. (2007) Near Linear Time Algorithm To Detect Community Structures in Large-Scale Networks. *Physical Review E*, **76**, Article ID: 036106. <https://doi.org/10.1103/PhysRevE.76.036106>