文章编号：1000-8608(2018)04-0414-08

基于标签传播的重叠社区发现算法

吴春国[1,2,3]，李艳振[1,2]，李 瑛[1,2]，高 瑞[1,2]，时小虎*[1,2,3]

（1.吉林大学 符号计算与知识工程教育部重点实验室，吉林 长春 130012；
2.吉林大学 计算机科学与技术学院，吉林 长春 130012；
3.吉林大学珠海学院 计算机学院，广东 珠海 519041）

摘要：重叠社区发现是复杂网络研究的重要课题.提出一种基于标签传播的重叠社区发现算法.首先利用标签传播算法得到初始无重叠社区划分结果，之后通过设计新的重叠节点识别算法确定重叠节点，最后再根据重叠节点的识别结果对社区进行合并从而得到最终的重叠社区划分结果.该算法克服了已有算法重叠节点占比过大的弊端.为验证算法的有效性，在LFR人工数据集、3个标准公开测试集以及真实的大豆基因共表达网络上进行实验，并与已有算法进行对比.实验结果表明，该算法性能明显优于对比算法，极大地改善了重叠节点比重过大问题.

关键词：重叠社区；社区发现；标签传播；复杂网络；基因表达数据

中图分类号：TP391　　**文献标识码**：A　　**doi**：10.7511/dllgxb201804012

0 引 言

现实世界中的许多问题都可以简化为复杂网络进行研究.因此，越来越多的研究者热衷于研究复杂网络，挖掘其隐藏价值.复杂网络拥有很多特性，如小世界性(small world)[1]、无标度性(scale-free)[2]和高聚集特性等.此外，复杂网络还呈现出明显的社区结构[3-4].社区结构也可称作网络簇结构.在社区结构中，同一个社区内边数很多，节点之间连接稠密，而跨社区相连的边相对比较稀疏.那些具有相似功能或属性的节点组成相同的社区，而不同的社区相互连接构成完整的复杂网络.相对于整个系统，社区就像人体结构中的一个器官或者组织，为了更好地解释和了解复杂系统功能，需要对社区的结构进行分析和研究.自从Girvan等[3]提出了社区发现的概念以来，复杂网络社区发现的研究成果不断涌现，主要可以分为谱方法、模块度优化方法、层次聚类方法和边预测方法等[5].

在实际生活中，复杂网络中的节点可以同时存在于多个社区中，因此这些社区之间会有重叠部分.在社交网络中，每个个体通常具有多个不同的社区属性，可能同时属于多个社会团体，如家庭、朋友圈、同事圈.同样，社区重叠现象广泛存在于生物分子网络中.比如在基因调控网络或者蛋白质相互作用网络中，单个基因或者蛋白质往往参与多个生物功能表达过程.因此，研究复杂网络的重叠社区或者重叠节点具有非常重要的意义.又比如在网络谣言的传播中，那些处于重叠节点位置的个体对于谣言的扩散传播起了决定性的作用.研究重叠节点的性质有助于深入理解谣言的传播机制.在生物网络中，不同的生物功能之间相互关联，并非完全割裂，重叠节点往往预示着关键信息，对于人类疾病的治疗、农作物抗病性的研究都具有重要意义.

近年来，重叠社区发现算法研究取得了很大进展，典型的算法大致可以分为5类[6]：派系过滤算法、边划分算法、局部扩展与优化算法、模糊发现算法、标签传播算法.Palla等于2005年提出了

收稿日期：2017-11-01；　**修回日期**：2018-03-15.
基金项目：国家自然科学基金资助项目(61373050)；吉林省科技发展计划青年科研基金资助项目(20130101070JC)；教育部在线教育研究中心在线教育研究基金资助项目(2017YB129).
作者简介：吴春国(1976-)，男，博士，副教授，E-mail：wucg@jlu.edu.cn；时小虎*(1974-)，男，博士，教授，博士生导师，E-mail：shixh@jlu.edu.cn.

派系过滤算法的代表方法——CPM(clique percolation method)[7]，其基本思想是复杂网络中多个派系(完全子图)之间相互重叠，构成了复杂网络中的社区. 派系过滤算法通过寻找相互连通 k-派系的方法确定社区结构. 派系过滤算法也可以实现重叠节点的社区发现，因为在派系过滤算法中的单个节点可能属于不同 k-派系. 边划分算法将复杂网络中的边进行划分，从而对复杂网络中的社区结构进行挖掘. 如果一个节点连接在一条边上，而且这条边被划分到多个边聚簇中，则此节点被判定为重叠节点[8-9]. 局部扩展与优化算法的社区发现过程是利用局部扩展与优化算法完成的. 在社区发现的过程中，这种算法使用局部社区或者已发现社区组成种子社区，而节点之间连接的紧密程度往往通过局部密度函数进行度量[10-11]. 模糊发现算法在每一个节点上计算归属因子向量(belonging factor)[12]，以此来计算社区与社区的联系强度和节点对的联系强度. 标签传播算法的主要思想是用标签传播的方式来确定每一个节点所属的社区，这是比较流行的一类算法. Gregory 等在非重叠节点社区发现的标签传播算法 LPA 基础上将每个节点的标签用多个类标签标识，并引入隶属度的概念，提出了 COPRA 算法进行重叠节点社区发现[13-14]. SLPA 算法也是一种标签传播的重叠社区发现算法，它通过模拟演讲者-收听者来完成标签传播过程[15]. 在 LPA 算法中，复杂网络中的节点不会记住在某一时刻接收到的标签信息. 与之不同的是，SLPA 算法会保存节点曾经收到的所有标签，为每一个节点设立一个用于存储对应标签概率分布的存储区，其中的存储概率分布表示了当前节点的归属强度. 文献[16]介绍了一种完全不一样的标签传播算法 SpeakEasy. 该算法运行速度快，适合不同种类的网络，但存在的问题是在识别重叠节点过程中可能出现重叠节点比重过大的现象.

本文借鉴 SpeakEasy 算法的思想，首先通过标签传播算法得到初始的无重叠社区划分，然后通过设计新的节点识别算法确定重叠节点，最后再对社区进行合并，提出 OCPLP(overlapping community partitioning based on label propagation)算法. 通过在 LFR 人工数据集、3 个标准公开测试集以及真实的大豆基因共表达网络上对本文提出的算法与已有算法进行对比.

1　SpeakEasy 算法简介

SpeakEasy 与 COPRA、SLPA 等算法一样，都是以标签传播为基础，通过标签传播把每个节点划分到对应的社区中. 不同的是，SpeakEasy 同时考虑整个网络的全局标签分布情况与局部标签分布情况，结合了自顶而下策略与自底而上策略对标签进行传播. 自顶而下策略主要是依据当前复杂网络中的全局标签分布情况来决定标签的传播，自底而上策略则主要考虑当前节点与邻居节点组成的局部子图中的标签分布信息.

SpeakEasy 算法可以分为两个阶段：第 1 个阶段是非重叠社区划分，首先进行标签传播过程，待标签传播收敛以后，可以提取到一个完整的非重叠社区划分结果，重复此过程 N_t 次，得到 N_t 个非重叠划分，从 N_t 个划分中筛选出一个最优划分. 如果不考虑重叠节点的话，此时已经得到了社区划分的结果. 第 2 个阶段是重叠节点识别，即在最优划分基础上识别重叠节点. 首先根据得到的 N_t 个划分结果计算共生矩阵 $\boldsymbol{A}$，$\boldsymbol{A}$ 中的元素 a_{ij} 表示节点 v_i 和 v_j 在 N_t 次划分中被聚为同一个社区的次数. 如果最优划分某个社区 $\boldsymbol{C}$ 之外的某个节点 v 与 $\boldsymbol{C}$ 中的节点的共生次数大于给定阈值，则可以认定 v 为 $\boldsymbol{C}$ 的重叠节点. 定义节点 v 和社区 $\boldsymbol{C}$ 的平均权值为

$$W_{v\boldsymbol{C}} = \frac{\sum_{u\in \boldsymbol{C}} a_{uv}}{|\boldsymbol{C}| \cdot N_t} \tag{1}$$

当 $W_{v\boldsymbol{C}}$ 大于给定阈值 γ 时，认定 v 为 $\boldsymbol{C}$ 的重叠节点.

SpeakEasy 算法的优势在于较少人工设定参数，适合不同种类的网络图，快速完成拥有大量节点的网络图的处理任务. 不足之处是此算法在识别重叠节点时会有重叠节点比重过大的现象.

2　OCPLP 算法

SpeakEasy 算法存在两个问题. 首先，当网络图规模较大且图中重叠节点较多时，两个社区之间会有大量的重合区域. 其次，小社区对大社区内的节点吸引力过大，会存在“蛇吞象”现象. 为了解决这两个问题，本文分别设计了新的重叠社区发现算法，并在最后增加了社区合并过程，提出了 OCPLP 算法，具体如下：

(1)随机初始化网络图

设整个网络图 $\boldsymbol{G}$ 包含 n 个节点，以每个节点的 ID 作为社区的标签信息. 首先为每个节点 i 建立大小为 N_{b} 的缓冲区，记为 b_i，用以保存最近 N_{b} 次更新的标签. 初始化时从该节点的邻居节点中随机抽取 N_{b} 次，将选中的邻居节点 ID 填入缓冲区，如图 1 所示.

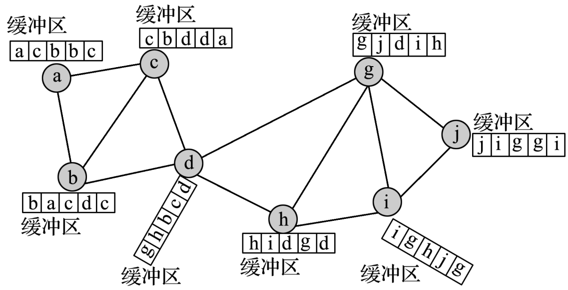


图 1　随机初始化网络图

Fig. 1　Random initialization of the network graph

(2)标签传播

①计算标签的全局概率分布，即计算所有标签在图 $\boldsymbol{G}$ 中全部节点缓冲区中的概率分布：

$$p_i=\frac{n_i}{N_{\mathrm{b}}\cdot n};\quad i=1,2,\cdots,n \tag{2}$$

其中 n_i 为第 i 个标签在图 $\boldsymbol{G}$ 中所有节点缓冲区出现的次数总和.

②计算每个节点邻居缓冲区的标签期望数. 设第 j 个节点的邻居节点数为 n_j^{n}，则其所有邻居节点缓冲区中第 i 个标签的期望数为

$$N_{ji}^{\mathrm{e}}=p_i\cdot n_j^{\mathrm{n}}\cdot N_{\mathrm{b}};\quad i,j=1,2,\cdots,n \tag{3}$$

③计算每个节点的标签局部特异性，即该节点的邻居节点缓冲区中标签的实际分布与期望分布之差. 记第 i 个标签在第 j 个节点的局部特异性为 s_{ji}，则其计算公式为

$$s_{ji}=N_{ji}^{\mathrm{a}}-N_{ji}^{\mathrm{e}};\quad i,j=1,2,\cdots,n \tag{4}$$

其中 N_{ji}^{a} 为第 i 个标签在第 j 个节点的所有邻居缓冲区中实际出现的个数.

④更新节点的缓冲区. 对于第 j 个节点的缓冲区 b_j，选择最大的 s_{ji} 所对应的标签作为该节点的新增标签，即删除 b_j 中的第 1 个元素，在队尾插入所选择的标签.

⑤重复②～④，遍历图 $\boldsymbol{G}$ 中所有节点.

⑥重复①～⑤，直到所有节点的缓冲区收敛.

例如在图 1 中一共有 8 个标签 a、b、c、d、g、h、i、j，其全局概率分布依次为 3/40、5/40、6/40、7/40、7/40、4/40、5/40、3/40. 以节点 d 为例，其邻居节点的缓冲区的标签 a、b、c、d、g、h、i、j 的实际数量分布为 2、2、3、6、2、2、2、1，总数为 20. 而按照全局概率分布，这 8 个标签的期望数分别为 1.5、2.5、3.0、3.5、3.5、2.0、2.5、1.5. 因此 8 个标签的特异程度分别为 0.5、−0.5、0、2.5、−1.5、0、−0.5、−0.5，最大的为标签 d 的 2.5. 因此对节点 d 的缓冲区进行更新时首先删除其第 1 个位置的 d，其余 4 个位置的 c、b、h、g 分别前移 1 位，末尾补充特异性最大的标签 d.

(3)抽取社区划分结果

根据上述得到的标签分布结果进行社区划分，具体过程如下：

①统计第 j 个节点所有邻居缓冲区中的标签数. 将数目最多的标签作为该节点的所属社区 ID. 该社区若已经存在，则将第 j 个节点划分到此社区中；若不存在，则以该 ID 新建社区，并添加第 j 个节点为该社区元素.

②重复①，遍历图中所有节点，假设共建立了 k 个社区，也就是说得到了一个包含 k 个社区的划分结果 $\boldsymbol{P}=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_k\}$.

(4)选择最优划分

重复(1)～(3) N_{t} 次，得到 N_{t} 个社区划分结果 $\{\boldsymbol{P}^{(1)},\boldsymbol{P}^{(2)},\cdots,\boldsymbol{P}^{(N_{\mathrm{t}})}\}$. 从中选择与其他划分最为一致的一次作为最优划分. 假设有 n 个节点，两次划分分别为 $\boldsymbol{P}^{(i)}=\{\boldsymbol{C}_1^{(i)},\boldsymbol{C}_2^{(i)},\cdots,\boldsymbol{C}_I^{(i)}\}$ 和 $\boldsymbol{P}^{(j)}=\{\boldsymbol{C}_1^{(j)},\boldsymbol{C}_2^{(j)},\cdots,\boldsymbol{C}_J^{(j)}\}$，$I$ 和 J 分别为 $\boldsymbol{P}^{(i)}$ 和 $\boldsymbol{P}^{(j)}$ 的社区个数. 采用调整排序指标(adjust rand index，ARI)作为评价两次划分的一致性指标，记 $\boldsymbol{P}^{(i)}$ 和 $\boldsymbol{P}^{(j)}$ 的调整排序指标为 R_{ij}，即有

$$R_{ij}=\left\{\sum_{s,t}\binom{n_{st}}{2}-\left[\sum_s\binom{n_{s.}}{2}\sum_t\binom{n_{.t}}{2}\right]\Big/\binom{n}{2}\right\}\Big/\left\{\frac{1}{2}\left[\sum_s\binom{n_{s.}}{2}+\sum_t\binom{n_{.t}}{2}\right]-\left[\sum_s\binom{n_{s.}}{2}\sum_t\binom{n_{.t}}{2}\right]\Big/\binom{n}{2}\right\} \tag{5}$$

其中 n_s 和 n_t 分别为 $\boldsymbol{C}_s^{(i)}$ 和 $\boldsymbol{C}_t^{(j)}$ 中的节点数，n_{st} 为既在 $\boldsymbol{C}_s^{(i)}$ 又在 $\boldsymbol{C}_t^{(j)}$ 中的节点数. 于是定义社区划分方案 $\boldsymbol{P}^{(i)}$ 的平均一致性为

$$\overline{R}_i=\sum_{j=1}^{N_{\mathrm{t}}}R_{ij}\Big/N_{\mathrm{t}} \tag{6}$$

选择最大评价一致性的划分为最优划分. 如果不

考虑重叠节点的话，该划分就是最终社区划分的结果.

(5)识别重叠社区节点

计算 N_t 个划分的共生矩阵 $\boldsymbol{A}$，元素 a_{ij} 表示节点 v_i 和 v_j 在 N_t 次划分中被聚为同一个社区的次数. 定义节点 v 和社区 $\boldsymbol{C}_i$ 的平均权值为

$$W_{v\boldsymbol{C}_i}=\frac{\sum\limits_{u\in \boldsymbol{C}_i} a_{uv}}{n\cdot \max\{|\boldsymbol{C}_i|,|\boldsymbol{C}_j|\}};\quad v\in \boldsymbol{C}_j \tag{7}$$

若 $W_{v\boldsymbol{C}_i}>\gamma_1$，则节点 v 为社区 $\boldsymbol{C}_i$ 的重叠节点，γ_1 为设定的阈值. 需要指出的是，式(7)中不仅考虑了社区 $\boldsymbol{C}_i$ 的规模，而且也考虑了社区 $\boldsymbol{C}_j$ 的规模，这样就在很大程度上避免了将大量大类节点计入小类的重叠节点，从而导致重叠节点比例过大的问题. 识别重叠社区节点算法的伪代码如下：

OCPLP-识别重叠社区节点

：　重叠节点的阈值 γ_1，共生矩阵 $\boldsymbol{A}$

：　最优划分 $\boldsymbol{C}$，图中全部节点集合 $\boldsymbol{G}$

function FindOverlapNodes(γ_1, $\boldsymbol{A}$, $\boldsymbol{G}$, $\boldsymbol{C}$)
　for $v\in\boldsymbol{G}$ do
　　for $c_i\in\boldsymbol{C}$ do
　　　$W_{v\boldsymbol{C}_i}=\frac{\sum\limits_{u\in \boldsymbol{C}_i} a_{uv}}{n\cdot \max\{|\boldsymbol{C}_i|,|\boldsymbol{C}_j|\}};\ v\in\boldsymbol{C}_j$
　　　if $W_{v\boldsymbol{C}_i}>\gamma_1$ then
　　　　$\boldsymbol{C}_i\leftarrow\boldsymbol{C}_i\cup\{v\}$
　　　end if
　　end for
　end for
end function

(6)社区合并

如果两个社区之间重合部分占比达到设定阈值，则合并这两个社区，即

$$\text{if}\quad \frac{|\boldsymbol{C}_i\cap\boldsymbol{C}_j|}{|\boldsymbol{C}_j|}>\gamma_2\quad \text{then}\quad \boldsymbol{C}_i\leftarrow\boldsymbol{C}_i\cup\boldsymbol{C}_j$$

其中 γ_2 为设定的阈值. 算法伪代码描述如下：

OCPLP-社区合并算法

：　合并社区的阈值 γ_2，共生矩阵 $\boldsymbol{A}$

：　最优划分 $\boldsymbol{C}$，图中全部节点集合 $\boldsymbol{G}$

function MergeCommunities(γ_2, $\boldsymbol{A}$, $\boldsymbol{C}$, $\boldsymbol{G}$)
　for $\boldsymbol{C}_i\in\boldsymbol{C}$ do
　　for $\boldsymbol{C}_j\in\boldsymbol{C}$ and $\boldsymbol{C}_j\neq\boldsymbol{C}_i$ do
　　　if $|\boldsymbol{C}_i\cap\boldsymbol{C}_j|/|\boldsymbol{C}_j|>\gamma_2$ then
　　　　for $v\in\boldsymbol{C}_j$ do　　//合并 $\boldsymbol{C}_j$ 到 $\boldsymbol{C}_i$
　　　　　if $v\notin\boldsymbol{C}_i$ then
　　　　　　$\boldsymbol{C}_i\leftarrow\boldsymbol{C}_i\cup\{v\}$
　　　　　end if
　　　　end for
　　　　delete $\boldsymbol{C}_j$
　　　end if
　　end for
　end for
end function

3　实　验

为验证本文算法的有效性，共设计了 3 个实验. 首先利用 LFR benchmark 算法[17]生成虚拟的重叠网络，在人工数据集上将本文提出的 OCPLP 与 SLPA[15]、SpeakEasy[16]两种当前主流重叠社区划分算法进行对比. 在第 2 个实验中选择几种常用的公开标准测试集，比较 OCPLP 与两种比较算法的性能. 最后一个实验选择了实际的大豆基因共表达网络，分别使用 SpeakEasy 和 OCPLP 算法对基因共表达网络进行重叠社区划分，并且比较两种算法的结果.

3.1　LFR benchmark 数据集对比

LFR benchmark 引入网络度分布和社区大小分布的指数等参数来生成重叠网络，所生成的网络能够模拟现实网络中的重要性质[17]. LFR benchmark 中提供了多种参数以控制生成网络的拓扑结构. 本文利用 LFR benchmark 工具生成了 3 个人工网络图：LFR1、LFR2、LFR3. 表 1 列出了生成 3 个网络图时所使用的参数，各个参数的定义如下：N 为节点数，m 为边数，k 为平均度，$k_{\max}$ 为最大度，μ 为混合程度，n_{on} 为重叠节点数，n_{oc} 为每个重叠节点从属的社区个数.

表 1　生成 LFR benchmark 网络图的参数

Tab. 1　Parameters used to generate LFR benchmark network graph

数据集	N	m	k	$k_{\max}$	μ	n_{on}	n_{oc}
LFR1	1 000	7 810	35	50	0.3	20	4
LFR2	4 000	21 568	10	60	0.3	100	5
LFR3	4 000	21 720	10	65	0.3	200	5

复杂网络重叠社区发现算法的性能常用模块度 Q[18]作为评价指标，其定义如下：

$$Q=\frac{1}{2m}\sum_{i,j}\frac{1}{O_iO_j}\left(N_{ij}-\frac{k_ik_j}{2m}\right)\delta(l_i,l_j) \tag{8}$$

式中：m 为网络中总的边数；O_i 表示节点 i 所属社区个数；$N_{ij}=1$ 代表节点 i 和节点 j 之间存在连边，否则不存在连边；k_i 为节点 i 度数；l_i 为节点 i 属于某个社区的标号；$\delta(l_i,l_j)=1$ 当且仅当 $l_i=l_j$.

评价复杂网络重叠社区发现算法的性能的另一个常用指标为标准化互信息，对两个划分 $\boldsymbol{P}^{(i)}$ 和 $\boldsymbol{P}^{(j)}$，其标准化互信息为[11]

$$I_{\mathrm{n}}(\boldsymbol{P}^{(i)},\boldsymbol{P}^{(j)})=\left[2\sum_{\substack{\boldsymbol{C}_s^{(i)}\in\boldsymbol{P}^{(i)}\\ \boldsymbol{C}_t^{(j)}\in\boldsymbol{P}^{(j)}}}\frac{|\boldsymbol{C}_s^{(i)}\cap\boldsymbol{C}_t^{(j)}|}{n}\times \log\frac{n\cdot|\boldsymbol{C}_s^{(i)}\cap\boldsymbol{C}_t^{(j)}|}{|\boldsymbol{C}_s^{(i)}|\cdot|\boldsymbol{C}_t^{(j)}|}\right]\Big/ \left[-\sum_{\boldsymbol{C}_s^{(i)}\in\boldsymbol{P}^{(i)}}\frac{|\boldsymbol{C}_s^{(i)}|}{n}\log\frac{|\boldsymbol{C}_s^{(i)}|}{n}-\sum_{\boldsymbol{C}_t^{(j)}\in\boldsymbol{P}^{(j)}}\frac{|\boldsymbol{C}_t^{(j)}|}{n}\log\frac{|\boldsymbol{C}_t^{(j)}|}{n}\right] \tag{9}$$

OCPLP 与两种比较算法的模块度 Q 和标准化互信息 I_{n} 的对比结果如表 2 所示. 缓冲区大小对结果整体影响不大，在本文实验中该参数取值为 5. 从表 2 可以看出，在人工数据集的两个指标上，OCPLP 算法比 SLPA、SpeakEasy 两种算法表现略好.

表 2　LFR benchmark 数据集上的对比结果

Tab. 2　Comparison results on LFR benchmark dataset

数据集	算法	Q	I_{n}
LFR1	SLPA	0.322 7	0.954 1
	SpeakEasy	0.311 9	0.950 9
	OCPLP	0.322 8	0.962 3
LFR2	SLPA	0.323 6	0.957 3
	SpeakEasy	0.327 3	0.971 6
	OCPLP	0.316 6	0.980 9
LFR3	SLPA	0.310 8	0.942 7
	SpeakEasy	0.308 9	0.968 3
	OCPLP	0.316 6	0.973 2

下面重点考察对重叠节点的识别情况，将识别重叠节点的过程理解为二分类问题，将重叠节点理解为正样本，将非重叠节点理解为负样本，用召回率 γ_{r}、精确率 γ_{p}、F_1 度量 3 个评估指标来评价 OCPLP 算法与两种比较算法的优劣.

$$\gamma_{\mathrm{r}}=n_{\mathrm{T}}/(n_{\mathrm{T}}+n_{\mathrm{N}}) \tag{10}$$

$$\gamma_{\mathrm{p}}=n_{\mathrm{T}}/(n_{\mathrm{T}}+n_{\mathrm{P}}) \tag{11}$$

$$F_1=2\times(\gamma_{\mathrm{r}}\times\gamma_{\mathrm{p}})/(\gamma_{\mathrm{r}}+\gamma_{\mathrm{p}}) \tag{12}$$

其中 n_{T} 为真阳性样本数，即对正类样本预测正确的样本数；n_{P} 为假阳性样本数，即把负类样本预测为正类的样本数；n_{N} 为假阴性样本数，即把正类样本预测为负类的样本数.

表 3 给出了在 LFR1、LFR2、LFR3 人工数据集上重叠节点识别的比较结果. 由表 3 可以看出，在 LFR1 上，3 种算法在 3 个指标的综合表现差异不大，其中 OCPLP 的表现最好. 在 LFR2 和 LFR3 上，3 种算法的表现差异明显，其中 SLPA 表现最差，而 OCPLP 的表现明显优于其他两种比较算法. 在 LFR1、LFR2、LFR3 人工数据集上的平均精确率，OCPLP 分别比 SLPA 和 SpeakEasy 提高了 83% 和 42%，而平均召回率则分别提高了 55% 和 22%，F_1 度量分别提高了 84% 和 40%. 可以看出，OCPLP 算法在这 3 个指标上全面优于 SpeakEasy 算法，而 SpeakEasy 算法又明显强于 SLPA 算法.

表 3　重叠节点识别的对比结果

Tab. 3　Comparison results of overlapped nodes identification

数据集	γ_{p}		
	SLPA	SpeakEasy	OCPLP
LFR1	0.90	1.00	1.00
LFR2	0.15	0.22	0.58
LFR3	0.27	0.49	0.84
数据集	γ_{r}		
	SLPA	SpeakEasy	OCPLP
LFR1	0.45	0.40	0.50
LFR2	0.42	0.39	0.49
LFR3	0.22	0.60	0.70
数据集	F_1		
	SLPA	SpeakEasy	OCPLP
LFR1	0.60	0.57	0.66
LFR2	0.22	0.28	0.53
LFR3	0.24	0.54	0.76

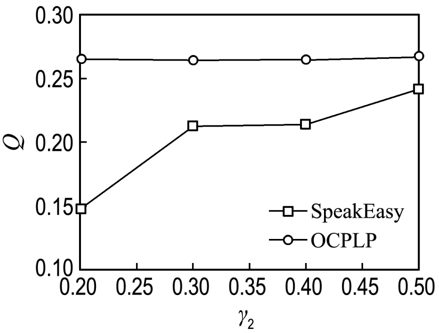
3.2　公开标准测试集对比

本文选择 pol. books[19]、arxiv 广义相对论学者合作网络(general relativity and quantum cosmology collaboration network)[20] 和 netscience[19] 3 个较为流行的公开数据集进行对比实验.

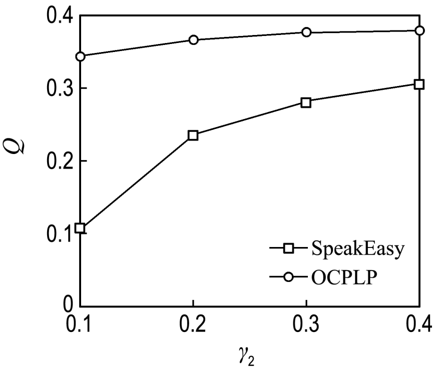
pol. books 是基于亚马逊网站的美国政治类

型书籍购买信息而构造的网络，有 105 个节点，441 条边；arxiv 广义相对论学者合作网络包括 5 242个节点和 28 980 条边；netscience 是复杂网络学者合作网络，由 1 461 个节点和 2 742 条边构成. 在 3.1 节的实验中可以看出 SpeakEasy 算法明显优于 SLPA 算法，所以在后面的实验部分只选择 SpeakEasy 算法与 OCPLP 算法进行对比.

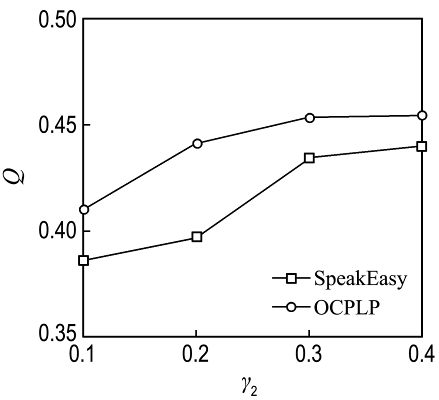
图 2 给出了 OCPLP 算法和 SpeakEasy 算法在 3 个数据集上的对比结果. 从图中可以看出，在 3 个真实数据集上 OCPLP 算法在不同阈值下的模块度 Q 都明显高于 SpeakEasy 算法. 其中，在



(a) pol. books



(b) arxiv 广义相对论学者合作网络



(c) netscience

图 2　典型网络数据集的对比结果

Fig. 2　Comparison results on the classical network datasets

pol. books 数据集上平均提高了 34.53%，在 arxiv 广义相对论学者合作网络上平均提高了 84.16%，而在 netscience 网络上平均提高了 6.30%.

3.3　大豆基因共表达网络对比实验

为了进一步验证所提算法的有效性，本文利用大豆基因共表达网络构造了社区发现算法的测试算例. 实验数据源于 GEO 数据库 GPL4592 平台下的 6 组大豆锈病相关的数据(GSE7108[21]、GSE8432[22]、GSE29740[23]、GSE29741、GSE33410[24]、GSE41724). 通过计算基因之间的皮尔森相关系数构建了一个大豆基因共表达网络. 该网络包含 4 169 个基因，21 135 条边，每两个基因之间的相似度作为对应边的权重，平均度为 10，平均聚类系数为 0.56. 针对所构建的大豆基因共表达网络，分别采用 SpeakEasy 算法和 OCPLP 算法对该网络进行社区划分.

从图 3 可以看出，OCPLP 算法在不同阈值下得到的模块度 Q 都较高，社区划分结果更好. 图 4

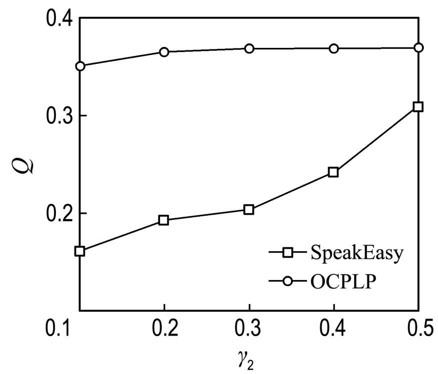


图 3　大豆基因共表达网络对比结果

Fig. 3　Comparison results on soybean gene co-expression network

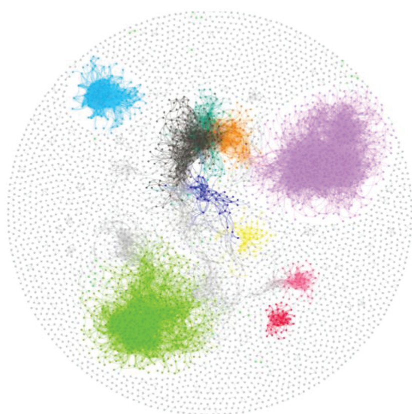


图 4　大豆基因共表达网络实验可视化效果

Fig. 4　Visual effect of soybean gene co-expression network experiment

给出了 OCPLP 算法对大豆基因共表达网络进行社区划分的可视化效果. 对社区划分结果做进一步分析有助于研究在锈病环境下大豆的基因共表达现象,并为大豆育种提供帮助.

4 结 语

针对重叠节点社区发现问题,本文通过设计新的重叠社区发现算法,增加社区合并过程,提出了 OCPLP 算法. 为验证所提算法的有效性,分别针对 LFR benchmark 人工数据集、3 个典型标准数据集以及实际的大豆基因共表达网络设计了 3 个实验,将本文提出的算法与现有算法进行了对比. 实验结果表明,本文提出的 OCPLP 算法性能明显优于对比算法,并极大改善了重叠节点比重过大的问题,使得结果更加符合问题的实际特征,也验证了 OCPLP 算法的有效性.

参考文献:

[1] WATTS D J, STROGATZ S H. Collective dynamics of 'small-world' networks [J]. **Nature**, 1998, **393**(6684):440-442.

[2] ADAMIC L A, HUBERMAN B A, BARABÁSI A L, *et al*. Power-law distribution of the World Wide Web [J]. **Science**, 2000, **287**(5461):2115.

[3] 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.

[4] RADICCHI F, CASTELLANO C, CECCONI F, *et al*. Defining and identifying communities in networks [J]. **Proceedings of the National Academy of Sciences of the United States of America**, 2004, **101**(9):2658-2663.

[5] 杨 博,刘大有,LIU Jiming,等. 复杂网络聚类方法[J]. 软件学报, 2009, **20**(1):54-66.
YANG Bo, LIU Dayou, LIU Jiming, *et al*. Complex network clustering algorithms [J]. **Journal of Software**, 2009, **20**(1):54-66. (in Chinese)

[6] XIE Jierui, KELLEY S, SZYMANSKI B K. Overlapping community detection in networks: The state-of-the-art and comparative study [J]. **ACM Computing Surveys**, 2013, **45**(4):43.

[7] PALLA G, DERÉNYI I, FARKAS I, *et al*. Uncovering the overlapping community structure of complex networks in nature and society [J]. **Nature**, 2005, **435**(7043):814-818.

[8] AHN Y Y, BAGROW J P, LEHMANN S. Link communities reveal multiscale complexity in networks [J]. **Nature**, 2010, **466**(7307):761-764.

[9] EVANS T S, LAMBIOTTE R. Line graphs, link partitions, and overlapping communities [J]. **Physical Review E — Statistical, Nonlinear, and Soft Matter Physics**, 2009, **80**(1):016105.

[10] BAUMES J, GOLDBERG M, KRISHNAMOORTHY M, *et al*. Finding communities by clustering a graph into overlapping subgraphs [C] // **IADIS International Conference on Applied Computing 2005**. Algarve: IADIS, 2005: 97-104.

[11] LANCICHINETTI A, FORTUNATO S, KERTÉSZ J. Detecting the overlapping and hierarchical community structure in complex networks [J]. **New Journal of Physics**, 2009, **11**(3):033015.

[12] ESQUIVEL A V, ROSVALL M. Compression of flow can reveal overlapping-module organization in networks [J]. **Physical Review X**, 2011, **1**(2): 021025.

[13] GREGORY S. Finding overlapping communities in networks by label propagation [J]. **New Journal of Physics**, 2010, **12**(10):103018.

[14] 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.

[15] XIE Jierui, SZYMANSKI B K, LIU Xiaoming. SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process [C] // **Proceedings — 11th IEEE International Conference on Data Mining Workshops, ICDMW 2011**. Piscataway: IEEE, 2011:6137400.

[16] GAITERI C, CHEN Mingming, SZYMANSKI B, *et al*. Identifying robust communities and multi-community nodes by combining top-down and bottom-up approaches to clustering [J]. **Scientific Reports**, 2015, **5**:16361.

[17] LANCICHINETTI A, FORTUNATO S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities [J]. **Physical Review E — Statistical, Nonlinear, and Soft Matter Physics**,

2009, **80**(1):016118.

[18] SHEN Huawei, CHENG Xueqi, CAI Kai, *et al*. Detect overlapping and hierarchical community structure in networks [J]. **Physica A: Statistical Mechanics and Its Applications**, 2009, **388**(8): 1706-1712.

[19] ROSSI R A, AHMED N K. The network data repository with interactive graph analytics and visualization [C] // **Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI 2015 and the 27th Innovative Applications of Artificial Intelligence Conference, IAAI 2015**. Austin: AI Access Foundation, 2015:4292-4293.

[20] LESKOVEC J, KLEINBERG J, FALOUTSOS C. Graph evolution: Densification and shrinking diameters [J]. **ACM Transactions on Knowledge Discovery from Data**, 2006, **1**(1):1217301.

[21] PANTHEE D R, YUAN J S, WRIGHT D L, *et al*. Gene expression analysis in soybean in response to the causal agent of Asian soybean rust (*Phakopsora pachyrhizi Sydow*) in an early growth stage [J]. **Functional & Integrative Genomics**, 2007, **7**(4):291-301.

[22] GREGORY A W, ROAYAEI J A, QUIÑONES O A. A microarray analysis for differential gene expression in the soybean genome using Bioconductor and R [J]. **Briefings in Bioinformatics**, 2007, **8**(6):415-431.

[23] SCHNEIDER K T, VAN DE MORTEL M, BANCROFT T J, *et al*. Biphasic gene expression changes elicited by *phakopsora pachyrhizi* in soybean correlate with fungal penetration and haustoria formation [J]. **Plant Physiology**, 2011, **157**(1):355-371.

[24] VAN DE MORTEL M, RECKNOR J C, GRAHAM M A, *et al*. Distinct biphasic mRNA changes in response to Asian soybean rust infection [J]. **Molecular Plant-Microbe Interactions: MPMI**, 2007, **20**(8):887-899.

An overlapping community identification algorithm based on label propagation

WU Chunguo[1,2,3], LI Yanzhen[1,2], LI Ying[1,2], GAO Rui[1,2], SHI Xiaohu*[1,2,3]

(1. Symbol Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China;
2. College of Computer Science and Technology, Jilin University, Changchun 130012, China;
3. School of Computer, Zhuhai College of Jilin University, Zhuhai 519041, China)

Abstract: Overlapping community identification is an important problem in complex network study. An overlapping community indentification algorithm based on label propagation is proposed. Firstly, label propagation algorithm is used to achieve the initial non-overlapping community structure. And then, new overlapping node detection algorithm is proposed to identify overlapping nodes. At last, according to the identification results of overlapping nodes, the communities are merged to get the final result of overlapping community partition. The proposed algorithm overcomes the disadvantages of the oversize overlapped nodes in existing algorithms. To verify the effectiveness of the algorithm, the experiments and comparison with existing algorithms are carried out on LFR artificial datasets, three benchmark open test datasets, and real soybean gene co-expression networks. The experimental result shows that this algorithm is clear superior to the existing algorithms, overwhelmingly improves the problem of great proportion of overlapping nodes.

Key words: overlapping communities; community identification; label propagation; complex networks; gene expression data