

# 一种基于聚集系数的社区发现算法<sup>1</sup>

樊梦佳, 张仰森

(北京信息科技大学, 北京市 100192)

**摘要:** 社区划分一直是复杂网络研究中的一个热门话题, 社区的快速准确划分为研究复杂网络的性质提供了良好的基础。传统的社区发现方法都是在全局复杂网络的基础上进行社区划分, 随着网络中节点的增加, 网络规模的变大, 使社区发现变得更为复杂。该文提出了一种局部社区发现算法, 该算法无需知道整个复杂网络的全部信息, 只需从一个待求节点出发, 考察其与邻接节点的紧密程度, 逐步将邻接点添加到社区中, 得到该节点所在的社区结构。同时该算法还可实现全局网络的社区发现。利用该算法分别对 Zachary 空手道俱乐部网络和海豚社会网络进行社区发现, 实验结果证明了该算法准确性与可行性。

**关键字:** 局部社区; 社区发现; 聚集系数

中图分类号: TP311

文献标识码: A

## Community Detecting Method Based on the Clustering Coefficient

Mengjia Fan, Yangsen Zhang

(Beijing Information Science and Technology University, Beijing 100192, China)

**Abstract:** Community detection has been a hot topic in the complex network research, the fast and accurate community detection can provide a good foundation for the research of complex network nature. Traditional community detection methods are on the basis of the global community, with the increase of nodes in the network, the size of network becomes larger, so that the community detection becomes more complex. The paper proposed a local community detection method, this method does not need to know the whole complex network information, it just starting from an initial node and calculating the tightness between the initial node and the adjacent nodes. And gradually add the adjacent nodes to the community; finally get this node community structure. Meanwhile, this method can achieve global network community detection. Applied the method to the Zachary karate club network and dolphin social network, and the experimental results demonstrate this method accuracy and feasibility.

**Keywords:** local community; community detection; clustering coefficient

### 1 引言

现实世界中的许多复杂系统都可以用复杂网络加以表示, 例如社会网络、生物网络、电力网络、web 网络等。通过两个简单的映射, 即对象到网络节点的映射与对象间关系到边的映射, 可将复杂网络表示成图模型。这种可以用图模式表示的复杂网络中除了拥有广为人知的小世界与无标度等特性外, 还具有极为重要的模块性, 即复杂网络中隐含着丰富的社区结构模式<sup>[1]</sup>。所谓社区, 可以松散地将其定义为具有某种共同特征的相互连接的信息载体集合。

针对如何准确快速地发现复杂网络中的社区结构一直是研究的热点, 中外研究学者们相继提出了多种社区发现算法。多种社区发现算法主要可以分为两大类, 第一类是在全局复杂网络的基础上进行社区发现, 第二类是无需知道网络的整体信息, 只需要从一个待求节点出发, 只考虑这个节点属于哪一个社区, 此种方法称为局部社区发现。G-N 算法<sup>[2]</sup>是一个比较经典的全局社区发现算法, 算法最初是由 Michelle Girvan 和 Mark Newman 提出的, 该算法属于分裂的层次聚类算法。其基本思想是不断删除网络中边介数最大的边, 然后重新计算网络中剩余边的边介数, 重复这个过程, 直到网络中所有的边都被删除。其不足之处在于在不知道社区数目的情况下, G-N 算法不知道这个分解要进行到哪一步结束为止。为解决这个问题, Newman 等人引进了一个衡量网络划分质量的标准——模块度, 模块度很好地解决了这个问题。针对此算法, 胡健等人<sup>[3]</sup>提出了基于边聚集系数的社区发现方法, 该方法类似于 G-N 算法, 其基本思想也是删除网络中的边, 但是此算法是删除网络中边聚集系数最小的边, 直到不存在满足定义的社区则停止算法。类似的针对全局网络的社区发现方法还有很多, 如谱平分算法、基于信息论法、基于标号传播和基于模糊聚类等的社区发现算法, 这些针对全局网络的社区发现算法往往会由于网络中节点的增多, 网络规模的变大而变得困难重重。同时, 在很多情况下, 我们往往只需要关心某个节点属于哪个社区, 在这种情况下我们就不需进行全局网络的社区发现, 只需要进行探索该节点属于哪个局部社区即可。

\* 收稿日期: 定稿日期:

基金项目: 国家自然科学基金项目(61370139); 北京市属高等学校创新团队建设与教师职业发展计划项目(IDHT20130519); 北京市教委专项(PXM2013\_014224\_000042、PXM2014\_014224\_000067)

所以针对这种需要，局部社区发现方法应运而生，本文提出了一种基于聚集系数的局部社区划分方法，该算法既可应用到全局网络社区发现中，又可实现局部网络社区发现。如果全局网络规模很大，则首先根据网络中边聚集系数值将网络划分为几个小网络，之后在网络内找到聚集度最高的节点作为初始节点，考察初始节点和其邻接节点之间的紧密程度，将邻接节点添加到社团中，重复以上步骤，直到所有节点都添加到社团中，即可得到全局网络的社区结构。如果针对局部网络进行社区发现，即可从一个待求节点出发，考察该待求节点与其邻接节点之间的紧密程度，将邻接节点添加到社区中，得到该节点所在的社区结构则停止算法，实现局部网络社区发现。

## 2 相关概念介绍

用  $G=\langle V,E \rangle$  表示一个无向无权网络，其中  $V$  表示节点集合， $E$  表示边集合。这个网络共有  $n$  个节点和  $m$  条边，用  $e_{ij}$  表示节点  $i$  和节点  $j$  之间存在的边，并且  $V=\{i|i \in n\}$ ， $E=\{e_{ij}|i,j \in n \text{ 且 } i \neq j\}$ 。

### A. 强连接社团

强连接社团定义为  $k_i^{in}(\delta) > k_i^{out}(\delta)$ ，其中， $\forall i \in V$ 。

其中， $\delta$  表示一个社团， $i$  表示社团中的任意一个节点， $k_i$  表示节点  $i$  的度，则  $k_i$  的值一部分来源于社团  $\delta$  内部，用  $k_i^{in}(\delta)$  表示；另一部分来源于社团  $\delta$  外部，用  $k_i^{out}(\delta)$  表示。则节点  $i$  的度  $k_i(\delta) = k_i^{in}(\delta) + k_i^{out}(\delta)$ 。所以对于强连接社团  $\delta$  来说，需要满足社团内的每一个节点  $k_i^{in}(\delta)$  均大于  $k_i^{out}(\delta)$ ，即  $k_i^{in}(\delta) > k_i^{out}(\delta)$ 。

### B. 弱连接社团

弱连接社团定义为  $\sum_{i \in V} k_i^{in}(\delta) > \sum_{i \in V} k_i^{out}(\delta)$ ；其中， $\forall i \in V$ 。

对于弱连接社团  $\delta$  来说，需要满足社团内的所有结点的  $k_i^{in}(\delta)$  之和大于  $k_i^{out}(\delta)$  的和，即  $\sum_{i \in V} k_i^{in}(\delta) > \sum_{i \in V} k_i^{out}(\delta)$ 。

### C. 聚集系数

#### a) 点聚集系数

节点的聚集系数定义为  $c(i) = \frac{E(i)}{T(i)}$ ， $C(i) \in [0,1]$ 。

其中， $i$  表示一个节点， $k_i$  表示节点  $i$  的度， $E(i)$  表示节点  $i$  的邻接节点之间的实际连接边数； $T(i)$  表示节点  $i$  的邻接节点之间可能形成的最大连接边数， $T(i)$  可以用节点的度  $k_i$  来表示， $T(i) = \frac{k_i * (k_i - 1)}{2}$ ， $C(i) \in [0,1]$ 。所以节点的聚集系数定义为节点  $i$  相连的  $n$  个邻接节点之间实际存在的连接边数与可能形成的最大连接边数的比值。点聚集系数体现了某节点的所有邻接节点之间连接的紧密程度。

#### b) 社团聚集系数

社团的聚集系数定义为  $c(\delta) = \frac{\sum_{i=1}^n c(i)}{n}$ ， $C(\delta) \in [0,1]$ 。

其中， $i$  表示社团中的一个节点， $n$  表示社团中节点的总数，社团聚集系数定义为社团内所有节点聚集系数的平均值。

#### c) 边聚集系数

边聚集系数定义为  $c(ij) = \frac{z_{ij}}{\min(k_i - 1, k_j - 1)}$ ， $C(ij) \in [0,1]$ 。

其中， $i$  和  $j$  分别表示社团中的一个节点，用  $e_{ij}$  表示节点  $i$  和  $j$  所连接的边， $z_{ij}$  表示网络中实际包含边  $e_{ij}$  的三角形的个数。分母为两个节点的度数值减 1 的最小值，表示网络中可能包含边  $e_{ij}$  的三角形的最大个数。所以边聚集系数定义为包含该边的实际存在的三角形个数与可能包含该边所形成的最大三角形个数的比值。边聚集系数体现了某条边两端节点之间连接的紧密程度。

### D. 邻居节点集合

#### a) 节点邻居节点集合

节点  $i$  的邻居节点集合  $N(i) = \{j | \text{节点 } i \text{ 和节点 } j \text{ 直接相连}\}$ 。

#### b) 社团邻居节点集合

社团  $\delta$  的邻居节点集合  $N(\delta) = \bigcup_{i=1}^n N(i)$ 。其中  $i$  表示社团内的一个节点， $n$  表示社团内共有  $n$  个节点。社团的邻居节点集合表示社团内所有节点的邻居节点集合的并集。

## 3 算法实现

### 3.1 算法思路

本文提出的社区发现算法既可以应用在全局网络社区发现中也可以应用在局部网络社区发现中。如果需要通过大规模全局网络的社区发现，则首先需要计算网络中所有边的边聚集系数，将边聚集系数为零的边进行删除，以此

将网络划分为几个小网络，这样可以减小复杂网络的规模，并且在进行后续的社区发现算法时，还可以降低算法的复杂度，提高算法效率；之后就可以在多个小网络中选取一个初始节点，并将这个节点添加到一个空社区中，形成一个局部社区；接着考察这个局部社区的所有邻接节点，设置判定条件，如果邻接节点满足判断条件，则将邻接节点添加到社区中，直到没有满足条件的邻接节点则停止判断；在剩余节点中继续选择节点作为初始节点，重复上述步骤，直到所有节点都添加到社区中，则停止算法。如果需要实现局部社区发现，则只需要从一个待求节点出发，根据判定条件考察待求节点和邻接节点之间的紧密程度，判断是否将邻接节点添加到社区中，最终得到待求节点所在的社区结构。

#### A. 初始节点的选取

在进行基于聚集系数的社区发现算法中，一般文章都会选择网络中度数最大的节点作为初始节点进行社区发现。这种选取方式方便简单，但是其只考虑到了网络中核心节点度数较大的特点，没有考虑到核心节点还有一个主要的特性就是其与邻接节点之间的联系比较紧密，并且其邻接节点之间也形成比较紧密的联系<sup>[4]</sup>。所以针对这个特点，文献[5]提出了聚集中心度概念，一个节点的聚集中心度定义为与该节点直接相连的边的边聚集系数之和，文章中选取边聚集系数之和最大的节点最为初始节点。此种选取方法只考虑了该节点与邻接节点之间的紧密程度，但是没有考虑到核心节点的所有邻接节点之间的联系也比较紧密的特点。由于节点的点聚集系数定义为节点邻接节点之间实际存在的边数与可能形成的最大边数的比值，反映了节点的邻接节点之间的连接紧密程度。所以本文在选取初始节点时，既考虑了节点与邻接节点之间的紧密程度，又综合考虑了邻接节点之间的紧密程度。所以定义了点聚集度的概念，点聚集度定义为节点的点聚集系数和与该节点直接相连的边聚集系数平均值之和，如公式 1 所示：

$$\text{Center}(i) = c(i) + \frac{\sum_{j \in N} c(ij)}{k_i} \quad \text{公式 1}$$

其中  $c(i)$  表示节点的点聚集系数， $c(ij)$  表示与节点  $i$  直接相连的节点  $j$  之间存在边的边聚集系数， $k_i$  表示节点的度数，则公式 1 的后半部分表示与节点  $i$  直接相连的所有边的边聚集系数的平均值。

#### B. 条件判定

##### a) 条件 1:

如果社区的任意一个邻接节点满足  $k_i^{\text{in}}(\delta) > k_i^{\text{out}}(\delta)$ 。

也就是说这个邻接节点有一半以上的连接在这个社区内，说明这个邻接节点与社区连接比较紧密，所以可以把这个邻接节点添加到社区中。

##### b) 条件 2:

如果一个社区的任意一个邻接节点的点聚集系数为 1，则将这个邻接节点和与其直接相连的其他节点均添加到这个社区中。

由于节点的聚集系数为 1 证明了这个节点所有邻接节点之间连接紧密，如果这个节点是一个社区的邻接节点，则说明这个社区和该节点的其他邻接节点之间连接紧密，即可划分到一个社区中。所以可将该节点和不包含在社区中的该节点的其他邻接节点添加到这个社区中。

##### c) 条件 3:

如果一个社区的聚集系数为 1，则将这个社区的所有邻接节点添加到这个社区中。

社区的聚集系数为 1，则证明这个社区中的所有节点的聚集系数为 1。由于节点的聚集系数为 1 证明节点的所有邻接节点之间相互联系紧密，并形成了全局耦合网络。所以社区的聚集系数为 1，就证明社区内的每一个节点都和其邻接节点之间的联系紧密，所以可以将这个社区的所有邻接节点均添加到这个社区中。

##### d) 条件 4:

如果社区内节点  $i$  和与其直接相连的节点  $j$  构成的边  $e_{ij}$  满足下列条件，则将这个节点  $j$  添加到社区内。如果边  $e_{ij}$  的边聚集系数值大于与  $j$  直接相连的所有点构成边的边聚集系数都大，则可将节点  $j$  添加到社区内。

由于边聚集系数体现了边两端节点的连接紧密程度，如果与某一节点相连的边聚集系数相对于与该节点相连的其他边的边聚集系数都大，则可以说明该边两端节点的连接比较紧密，则可将该边的另一个节点添加到社区中。

##### e) 条件 5:

社区内节点满足下列两种情况中的任何一种：1、 $k_i^{\text{in}}(\delta) > k_i^{\text{out}}(\delta)$ ；2、 $\sum_{i \in V} k_i^{\text{in}}(\delta) > \sum_{i \in V} k_i^{\text{out}}(\delta)$

如果社区内节点满足第一种情况，即社区内任意一个节点的内度均大于外度，则所形成社区为强连接社区；如果社区满足第二种情况，即社区内所有节点的内度和大于外度和，则所形成的社区为弱连接社区。

#### C. 社区的合并

根据上述算法，可以将网络划分为多个子社区，接下来我们要考察社区的性质，判断所形成的社区是否为正确的社区结构，如果不是正确的社区结构则需要进行社区的合并。根据条件 5 计算社区内所有节点的内度和外度，判

断每个社区是否为强连接社区或弱连接社区，如果不属于上述任何一种社区，我们在这里称为非连接社区，则将非连接社区和与其邻接的社区进行合并，合并流程图见图 1。

具体合并过程条件如下：

1、如果所形成的社区为强连接社区或弱连接社区，则不进行社区合并；

2、如果所形成的社区为非连接社区，则将这个社区和与其直接相连的弱连接社区或非连接社区进行合并，如果形成的新社区相对于合并之前的社区连接度有所提高（也就是非连接社区经过合并，形成了弱连接社区或强连接社区），则保留社区合并结果；如果连接度并未提高（也就是非连接社区经过合并之后，仍为非连接社区），则继续将非连接社区和与其直接相连的其他社区进行合并，直到得到连接度有所提高的社区或者没有可以合并的社区为止，则结束合并。

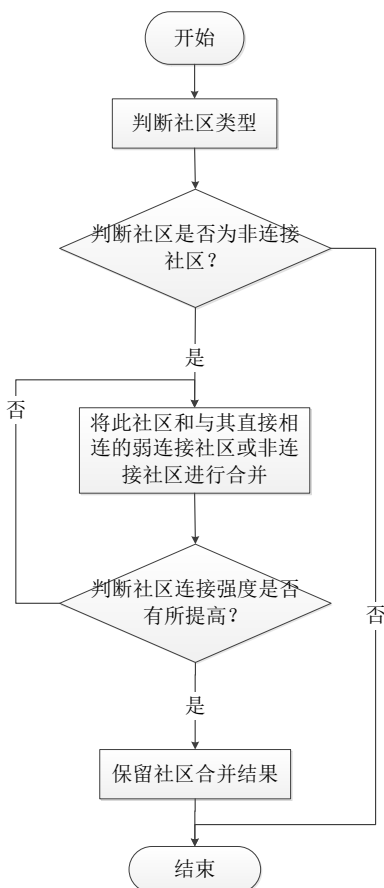


图 1 社区合并流程图

### 3.2 算法实现

算法具体实现步骤见算法 1：

算法 1 基于聚集系数的社区发现算法

**Step1:** if(在知道网络整体信息的基础上实现大规模全局社区发现)

计算网络中所有边的边聚集系数  $c(ij)$ ，存储到 Dictionary<string, double>类型的 edge 中

foreach (KeyValuePair<string, double> kv in edge)

if(kv.value==0)

edge.Remove(kv.Key)，删除  $c(ij)=0$  的边，将大规模网络划分为几个小网络，减小复杂网络规模，将每小社区存储到 Dictionary<int, string> 类型的 com 中

foreach (KeyValuePair<int, string> kv in com)

在每个小社区中计算所有节点聚集度  $Center(i)$ ，将每个节点的点聚集度存储到 Dictionary<int, double>

类型的 node 中

foreach (KeyValuePair<int, double> kv in node)

得到  $Max[Center(i)]$ 对应的节点  $i$  作为初始节点，goto Step3

**Step2:** if(进行局部网络社区发现)

从待求节点  $i$  出发，goto Step3

**Step3:** 将节点  $i$  添加到一个空社区中，形成 community[0]

将节点  $i$  的邻接节点集合存储到  $\text{Dictionary}\langle \text{int}, \text{List}\langle \text{int}\rangle\rangle$  类型的  $N$  中

循环  $\text{List}$  中的所有邻接节点

$\text{if}(\text{邻接节点满足 condition1} \mid \mid \text{condition2} \mid \mid \text{condition3} \mid \mid \text{condition4})$

将邻接节点添加到  $\text{community}[0]$  中, 对应的将  $\text{List}\langle \text{int}\rangle$  类型的  $\text{Nodes}$  节点集合中的节点进行删除

重复进行 4 个条件判断, 直到没有满足条件的节点, 社区结构不再变化为止, 则停止算法, 得到局部社区  $\text{community}[0]$ , 算法结束

$\text{if}(\text{进行全局社区发现})$

在所有  $\text{com}$  初步社区中重复进行初始节点的选取和 Step3, 直到  $\text{Nodes}$  集合为空, 也就是所有节点都添加到了相应社区中, 得到  $n$  个  $\text{community}$ ,  $\text{goto Step4}$

**Step4:** 计算  $n$  个  $\text{community}$  的社区连接度并存储到  $\text{Dictionary}\langle \text{int}, \text{int}\rangle$  类型的  $\text{nature}$  中, 非连接社区用 0 表示, 弱连接社区用 1 表示, 强连接社区用 2 表示

$\text{foreach}(\text{KeyValuePair}\langle \text{int}, \text{int}\rangle \text{ kv in nature})$

$\text{if}(\text{kv.value}==0)$

将此社区和邻接社区进行合并, 重新计算社区连接度

$\text{if}(\text{连接度}>0)$ , 也就是经过合并之后, 社区的连接度有所提高, 则保留社区合并结果, 相应的更新社区  $\text{community}$

$\text{if}(\text{连接度}==0)$ , 社区连接度没有提高, 重新将此社区和其他相连的邻接社区进行合并, 直到经过合并之后的社区连接度有所提高, 或是没有新的邻接社区可以进行合并, 则算法结束。得到最终的  $n$  个  $\text{community}$ , 得到社区划分结果

## 4 算法应用

### 4.1 Zachary 社会关系网

Zachary 空手道俱乐部网络是用来判断社区划分效果的常用测试网络<sup>[8]</sup>, 该网络是一个关于美国一所大学空手道俱乐部成员之间相互关系的网络。该网络共含有 34 个节点和 78 条边, 节点表示俱乐部中的成员, 边表示成员之间联系关系。由于俱乐部是否提高收费这一问题使俱乐部管理者 (节点 34) 和俱乐部老师 (节点 1) 之间发生冲突, 使俱乐部分裂成为两个小的社团。这两个小社团分别以节点 34 和节点 1 为核心, 如图 2 所示, 图中不同形状的点代表分裂后属于不同社区的节点。

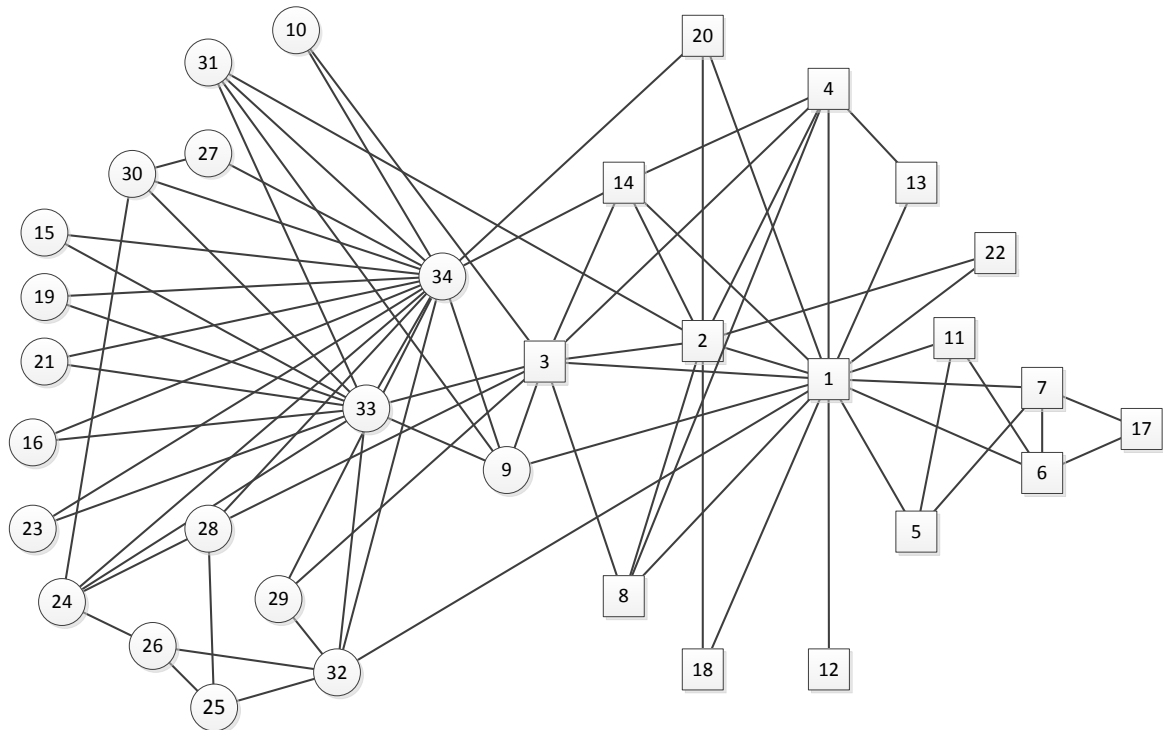


图 2 空手道社会关系网

利用本文算法, 在进行初步社区划分后, 将该关系网络划分为了三个小的社区, 如图 3 所示, 社区 1 内节点包括{1,2,3,4,5,8,11,12,13,14,18,20,22}; 社区 2 内节点包括{9,10,15,16,19,21,23,24,25,26,27,28,29,30,31,32,33,34}; 社区 3 内节点包括{6,7,17}。接着进行社区合并, 根据合并规则, 计算社区的性质, 社区 1 为弱连接社区, 社区 2 为强连接社区, 社区 3 为非连接社区。所以将社区 3 和其邻接的社区 2 进行合并, 得到一个新的社区 2, 判断社区性质为弱连接社区, 社区连接度有所提高, 保留最终社区合并结果, 社区最终结构和网络的真实结构相同, 如图 2 所示。

表 1 为使用本算法实现社区发现，和两个类似算法的比较。实验结果表明本文使用的算法，在初步社区划分后所形成的社区个数少，更贴近实际社区划分，三种算法的最终社区结构和真实网络相同。

表 1 空手道社会关系网社区划分实验比较

方法	文献[7]算法		文献[5]算法		本文算法	
	社区个数	社区成员	社区个数	社区成员	社区个数	社区成员
初步社区划分	5	{1,2,3,4,5,8,11,12,13,14,18,20,22}	5	{1,2,3,4,5,8,11,12,13,14,18,20,22}	3	{1,2,3,4,5,8,11,12,13,14,18,20,22}
		{9,15,16,19,21,23,31,33}		{9,15,16,19,21,23,24,27,28,30,31,33,34}		{9,10,15,16,19,21,23,24,25,26,27,28,29,30,31,32,33,34}
		{6,7,17}		{6,7,17}		{6,7,17}
		{24,25,26,28}		{25,26,29,32}		
		{10,27,29,30,32,34}		{10}		
最终社区结构	2	{1,2,3,4,5,6,7,8,11,12,13,14,17,18,20,22}	2	{1,2,3,4,5,6,7,8,11,12,13,14,17,18,20,22}	2	{1,2,3,4,5,6,7,8,11,12,13,14,17,18,20,22}
		{9,10,15,16,19,21,23,24,25,26,27,28,29,30,31,32,33,34}		{9,10,15,16,19,21,23,24,25,26,27,28,29,30,31,32,33,34}		{9,10,15,16,19,21,23,24,25,26,27,28,29,30,31,32,33,34}

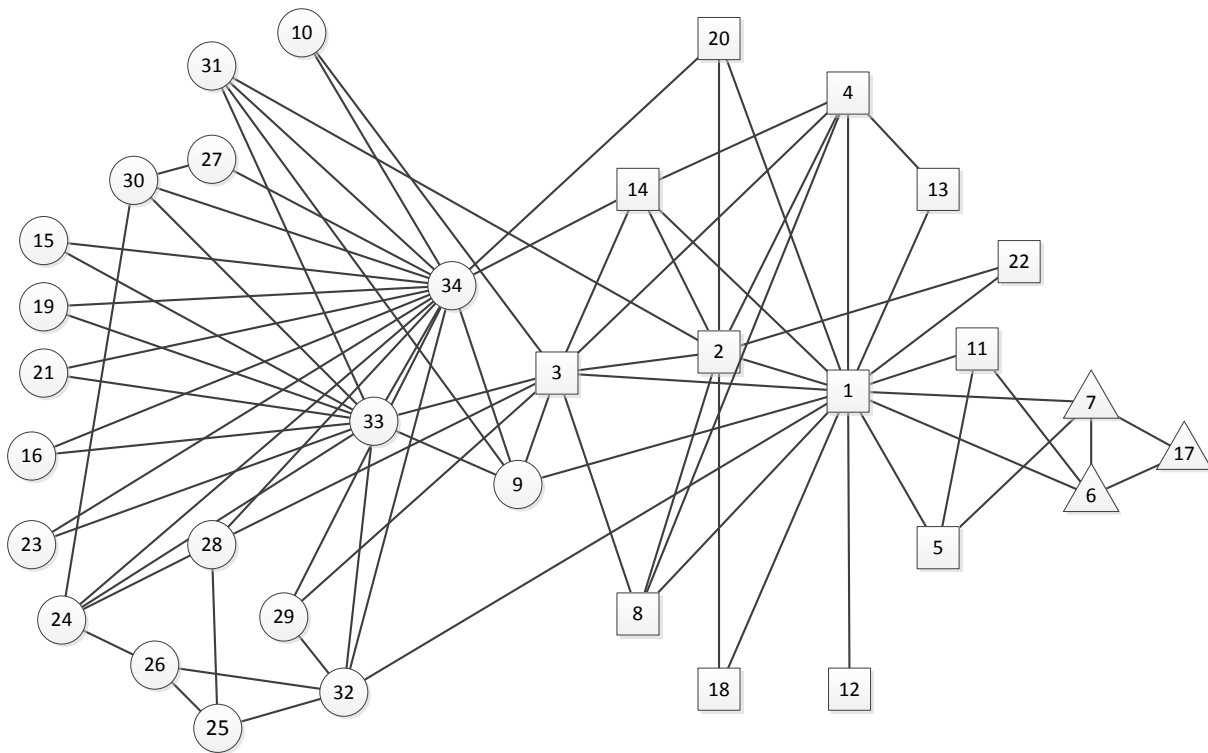


图 3 社区初步划分结果

#### 4.2 Dolphin social network 海豚关系网络

Dolphin social network 也是一个真实存在的网络<sup>[9,10]</sup>，也经常用于测试社区的划分效果。该网络是一个关于生活在新西兰中的 62 只海豚之间相互关系的网络。该网络是由 62 个节点和 159 条边构成的网络，节点表示海豚，边表示两个海豚之间的接触关系。该海豚网络是由两个海豚家族构成的，其中一个较大的海豚家族包括 42 个成员，较小的海豚家族包括 20 个节点，如图 4 所示，图中不同形状的节点代表不同家族的海豚成员。

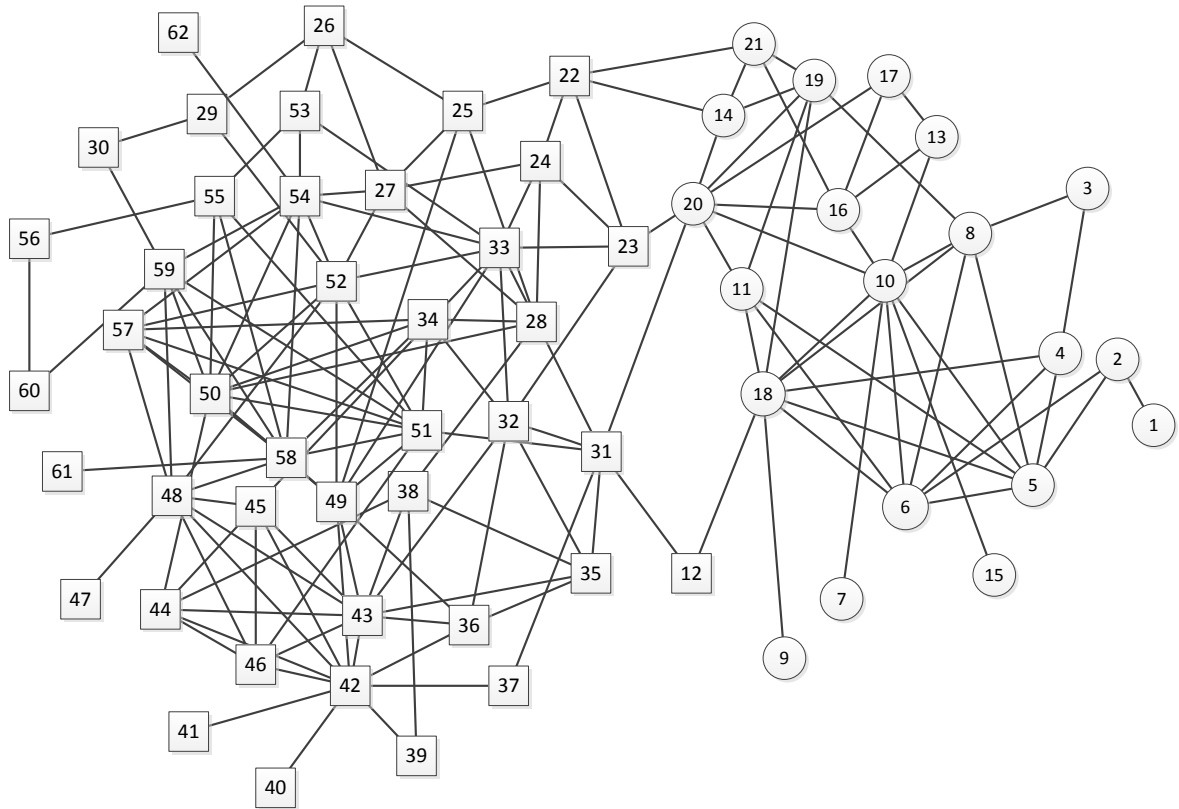


图 4 海豚关系网络

在对社区进行初步划分之后，文献[3]中的算法将网络初步划分为 30 个小社区，文献[2]的算法将网络划分为 14 个小社区，利用本文算法得到初步 11 个小社区。经过社区合并之后，文献[3]最终合并形成 3 个社区，和真实网络社区结构不符，文献[2]最终合并形成两个社区，社区个数和真实网络相同，但是节点 12 所属社区划分错误。利用本文算法进行社区合并之后，形成最终正确的两个社区，并且节点 12 所属社区划分正确。

表 2 海豚关系网络社区划分实验比较

方法	文献[6]算法		文献[5]算法		本文算法	
	社区个数	社区成员	社区个数	社区成员	社区个数	社区成员
最终社区划分	3	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21}	2	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21}	2	{1,2,3,4,5,6,7,8,9,10,11,13,14,15,16,17,18,19,20,21}
		{35,37,38,39,40,41,42,43,44,45,46,47,48}		{22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62}		{12,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62}
		{22,23,24,25,26,27,28,29,30,31,32,33,34,36,49,50,51,52,53,54,55,56,57,58,59,60,61,62}		{6,57,58,59,60,61,62}		

## 5 结束语

本文提出了一种基于聚集系数的局部社区发现算法，该算法既可应用于全局网络的社区发现，又可应用于局部网络的社区发现。在进行局部网络社区发现中，从一个待求节点出发，根据判定条件考察待求节点和邻接节点的紧密程度，将邻接节点添加到社区中，得到待求节点所在的社区结构。在进行全局网络社区发现中，如果网络规模很大，可以首先将复杂网络中边聚集系数为零的边进行删除，将复杂网络的规模降低。之后在每个小网络中计算每个节点的聚集度，选则聚集度最高的节点作为初始节点，并将这个节点添加到社区。然后根据判定条件，逐步考察初始节点与邻接点之间的紧密关系，逐步将联系紧密的节点添加到社区中，直到所有节点都成功划分到社区中。该算法在进行局部网络划分时，不用知道复杂网络的整体信息，就可以进行社区的划分，有效解决了由于节点增多，网络规模变大，社区划分更难的问题。并且一般全局网络社区划分在初始社区划分中，基于聚集系数的社区划分方法

都会将网络划分为多个小社区，再根据定义进行社区的合并，无疑增大了工作量。本文提出的全局网络社区发现算法在初步社区划分之后，形成的小社区相对其他算法来说比较少，社区不是很零散，进行社区合并比较简单方便，算法更为简洁准确。

## 参考文献

- [1]黄发良,张师超,朱晓峰.基于多目标优化的网络社区发现方法[J].软件学报,2013,24(9):2062-2077.
- [2] Newman M E J, Girvan M. Finding and evaluating community structure in networks[ J] , Phys Rev E , 2004 , 69 :026113.
- [3] 胡健, 杨炳儒. 基于边聚集系数的社区结构发现算法[J]. 计算机应用研究, 2009, 26(3):858-859.
- [4]王磊,吕强,李亚飞,等.一种基于个体中心论的局部优化社区发现算法[J].计算机应用与软件,2010,27(10):269-271.
- [5] 张睿. 基于点聚集系数和边聚集系数的社区发现算法[D]. 云南大学, 2013.
- [6] 王学凯, 马英红. 一种基于聚集系数的复杂网络社团划分算法[J]. 网络安全技术与应用, 2012 (9): 53-55.
- [7] 李孔文, 顾庆, 张尧, 等. 一种基于聚集系数的局部社团划分算法[J]. 计算机科学, 2010, 37(7): 46-49.
- [8] Zachary WW. An information flow model for conflict and fission in small groups. Journal of Anthropological Research,1977,33(4):452-473.
- [9] Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations—Can geographic isolation explain this unique trait? Behavioral Ecology and Sociobiology, 2003,54(4):396-405.
- [10] Lusseau D, Newman MEJ. Identifying the role that animals play in their social networks. Proc. of the Royal Society B: Biological Sciences, 2004,271(Suppl\_6):S477-S481.

**作者简介:** 樊梦佳 (1991-), 女, 硕士研究生, 主要研究领域为自然语言处理。Email: [fmj\\_bistu\\_edu@126.com](mailto:fmj_bistu_edu@126.com); 张仰森 (1962-), 男, 教授, 主要研究领域为自然语言处理、人工智能、Web 内容安全、智能立体仓储。Email: [zhangyangsen@163.com](mailto:zhangyangsen@163.com)。

