

A Community Detection Algorithm Based on Cooperative Game

Chao Cheng, Lihua Zhou*, Depin Yang, Kezhen Long, Caiping Zhang

Department of Computer Science and Engineering, Yunnan University, Kunming
Email: lhzhou@ynu.edu.cn

Received: Aug. 6th, 2014; revised: Sep. 16th, 2014; accepted: Sep. 30th, 2014

Copyright © 2014 by authors and Hans Publishers Inc.

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

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



Open Access

Abstract

With the rapid development of Internet technology, the virtual large-scale social networks are widespread. Community structure is an important characteristic of social network, mining the community structure in large-scale social networks can help us to understand the internal structure and relationships of the network, so as to better apply these networks. Therefore, community detection has important practical significance. A kind of individual rationality as the core of cooperative game community detection algorithm was proposed in this paper, which based on cooperative game community detection model and efficient iterative formula for computing the Shapley value (SH). CDCG algorithm includes initial detection and community adjustment. In the initial detection of changing the strategy environment, every node based on its own maximum SH value to make a decision, after several rounds of decisions, when all of the nodes' SH value become balance in the network, then the initial detection ended. The characteristics of the internal tight connection of community and relatively sparse in communities, can detect the unreasonable and meaningless small clusters in step of community adjusting, then the detected cluster has obvious community features. In order to improve efficiency of the algorithm, no contribution nodes pruning and ownership nodes pruning strategies were proposed. Finally, extensive experiments show that CDCG algorithm can automatically determine the final number of divided communities, which is effective and efficiency.

Keywords

The Social Network, Community Detection, Cooperative Game

一种基于合作博弈的社区检测算法

程超, 周丽华*, 杨德品, 龙克珍, 张彩平

*通讯作者。

云南大学信息学院计算机科学与工程系, 昆明
Email: lhzhou@ynu.edu.cn

收稿日期: 2014年8月6日; 修回日期: 2014年9月16日; 录用日期: 2014年9月30日

摘要

随着互联网技术的迅速发展, 虚拟大规模社会网络普遍存在。社区结构是社会网络的重要特征, 挖掘大规模社会网络中的社区结构能帮助人们了解网络中的内部结构和关系, 从而更好的应用这些网络。因此, 社区检测具有重要的现实意义。本文基于合作博弈检测社区的模型及高效计算Shapley值(SH值)的迭代公式, 提出了一种以个体理性为核心的合作博弈社区检测算法(CDCG算法)。CDCG算法包括初检测和社区调整两个步骤, 初检测中每个节点在不断变化的策略环境下依据自身获得最大SH值作出决策, 经过多轮决策后, 网络中所有节点的SH值达到平衡状态, 初检测结束; 社区调整是利用社区内部连接紧密、社区之间连接相对稀疏的特征对初检测产生的不合理、无意义的小簇进行调整, 使得检测得出的簇具有明显的社区特征。为了提高算法时间效率, 本文提出了无贡献节点剪枝策略和已归属节点剪枝策略。最后通过大量实验验证了CDCG算法能够自动确定最终社区划分个数且具有较好的社区检测效果及时间效率。

关键词

社会网络, 社区检测, 合作博弈

1. 引言

现实世界中的很多系统都可以用复杂网络模型来描述, 复杂网络成为了跨学科的研究热点和研究复杂系统的重要方法。社会网络是复杂网络中的一类, 反映了社会成员及其相互关系。随着互联网技术的迅猛发展, 人们的的生活方式呈现互联网交际、互联网消费、互联网金融或理财等特征, 许多的虚拟社会网络不断出现, 如社交网(人人网、Facebook)、微博(新浪微博、Twitter)、电子商务网(淘宝、京东、亚马逊)、微信平台等, 这些虚拟社会网络聚集了大量用户[1]。用户处在网络虚拟空间中, 却在真实的共享信息和知识、讨论热点事件、交流情感及购物感受、分享生活等, 他们是信息的浏览者和接受者, 也是互联网信息资源的提供者和传播者, 是商品的供应者和潜在的消费者。研究这些用户形成的复杂社会网络具有现实的意义及商业应用的价值, 而社会网络中用户的大量社会行为数据被保存在网络空间中, 为研究这些大规模社会网络创造了很好的条件。社会网络研究近些年受到了国内外研究者的广泛关注, 其中社会网络中社区结构研究更是被多个学科领域所重点关注, 并进行了深入探讨和研究, 取得了丰硕的成果。但由于社会网络是复杂的且规模庞大的, 所以社区检测仍然是有困难的, 人们依然在不断的创新并提出新的社区检测的算法。经典的社区检测算法一般是面向结构的分析方法, 但随着社会网络分析研究的不断深入, 更多的学者提出将社会网络作为一个智能网络系统, 网络中的节点作为智能的社会个体, 提出了一些基于社会主体(个体)行为的方法, 这样的方法分析出的社区结构, 是网络中的智能个体, 在不断选择和协调的过程中形成的, 符合真实社会网络的实质。本文提出的基于合作博弈的社区检测算法, 属于一种基于社会个体行为的方法。本文在文献[2]提出的合作博弈模型及高效计算 Shapley 值迭代公式的基础上, 展开研究工作, 提出了一种以个体理性为核心的合作博弈的社区检测算法, 算法是基于网络中个体在不断根据自身利益的抉择及协调的过程中实现社区的检测, 算法能够自动确定最终社区划分个数且具有较好的社区检测效果及时间效率。

2. 相关工作

社区检测一直是复杂网络研究领域的热点问题。随着人们对复杂网络中社区检测问题深入而广泛的研究, 研究者们提出了很多社区检测算法。其中 GN 算法[3]是目前进行网络社区检测的常用算法, 该算法在中小规模的网络中能较好地检测社区。

近年来, 研究者们提出了一些较新的算法, 如 Zhao[4]等人提出了基于主题为导向的社区划分方法; Shi [5]等提出了多目标社区检测的算法, 其实质是利用计算模块度的公式建立两个函数, 将社区检测问题转化了函数最优化问题; Tasgin [6] [7]等提出了利用遗传算法来完成社区检测, 达到了较好的结果; Ahn [8]等提出了边相似度划分社区的方法, 该方法能够找出重叠社区, 也称之为边社区; Chen [9]等人提出了利用社区形成博弈发现社区的算法, 其主要的思想是将社区动态形成过程描述成一个策略博弈过程(社区形成博弈), 并基于 Newman 模块度函数提出了一个增益函数和一个损耗函数来反映和衡量一个节点加入社区的得失, 在这样的博弈框架下一个社区被解释成一个博弈平衡, 由于一个节点可以选择多个社区, 因此 Chen 的算法可以发现重叠社区; Zhou [2]等人建立了合作博弈社区检测模型, 提出了高效计算 Shapley 值的迭代公式, 并设计了一种合作博弈的凝聚层次聚类算法(GAMEHC 算法), 该算法主要思想是联盟之间相互合作, 计算合作带给联盟的收益增量 ΔSH , 并每次取出增量最大的联盟之间的合作, 作为一次决策, 重复上述决策过程直到所有节点成为一个社区, 终止算法。该算法主要是基于合作博弈的方法而 Chen 等人采用的是策略博弈, 是一种非合作博弈的方法。合作博弈是一种强调团体理性的博弈, 且个体贡献的收益在团体中被最大化, 这符合实际社区以收益最大化形成社区是一致的。社区形成过程近似于合作博弈形成联盟的过程, 只是如何构建合作博弈模型, 如何分配收, 而非合作博弈尽管在实际社区中仍然存在, 但难以体现群体形成过程和原因。因此, 在文献[2]基础上提出本文基于合作博弈的社区检测算法。

3. 基于合作博弈的社区检测算法

本节将阐述本文合作博弈的社区检测算法, 首先介绍了相关定义及定理, 包括合作博弈模型的社区检测模型定义和高效计算 Shapley 值的迭代公式定理, 然后阐述了本文算法的初检测算法和社区调整算法, 最后给出了本文合作博弈的社区检测算法伪代码并分析了算法时间复杂度。

3.1. 相关定义及定理

定义 1 [2] 合作博弈的社区检测模型。 若 $N = \{1, 2, \dots, n\}$ 是复杂网络中所有节点的集合, $A = (a_{ij})_{n \times n}$, $i, j \in N$ 是复杂网络的邻接矩阵, 如果节点 i 与节点 j 之间存在边, 则 $a_{ij} = 1, i, j \in N$, 否则, $a_{ij} = 0, i, j \in N$, 则合作博弈的社区检测模型为 $CCG = (N, v)$, $i \in N$ 是联盟博弈参与节点。合作博弈的特征函数 $v(S)$ 定义为如下公式:

$$v(S) = \begin{cases} v(S) = 0, S \subseteq N, |S| = 1, \text{ or } S = \Phi \\ v(S) = \sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} \frac{a_{ij}}{d(i)}, S \subseteq N, |S| \geq 2, d(i) \neq 0 \end{cases} \quad (1)$$

其中, $d(i) = \sum_{j \in N} a_{ij}$ 为节点 i 的度, 若 $d(i) = 0$, 则 $\frac{a_{ij}}{d(i)} = 0$ 。 $\frac{a_{ij}}{d(i)} \geq 0$, 因此 $v(S) \geq 0$ 。 $v(S)$ 为 N 的子集(联盟)获得的“收益”。这个收益代表着联盟中所有关联的节点的边际贡献的总和。

定理 1 [2] 高效计算 Shapley 值公式。 给出社区检测合作博弈模型 $CCG = (N, v)$, $\forall S_1, S_2 \subseteq N$, $S = S_1 + S_2$, 则 $i \in S_1$ 和 $j \in S_2$ 在联盟 S 中的 Shapley 值计算公式如下:

$$\begin{aligned}
SH_v(S, i) &= SH_v(S_1, i) + \frac{1}{2} \sum_{j \in S_2} \left(\frac{a_{ij}}{d(i)} + \frac{a_{ji}}{d(j)} \right), i \in S_1 \\
SH_v(S, j) &= SH_v(S_2, j) + \frac{1}{2} \sum_{i \in S_1} \left(\frac{a_{ij}}{d(i)} + \frac{a_{ji}}{d(j)} \right), j \in S_2
\end{aligned} \tag{2}$$

其中, $d(i) = \sum_{j \in N} a_{ij}$ 为节点 i 的度, 若 $d(i) = 0$, 则 $\frac{a_{ij}}{d(i)} = 0$, $d(j), \frac{a_{ji}}{d(j)}$ 同理。

3.2. 初检测算法

合作博弈主要强调的是群体理性, 但这种群体理性依然是以个体理性为基础的。因此本文算法首先以个体理性为核心, 进行初步检测, 再以群体理性做社区调整。初检测算法主要思想是将网络中所有的节点作为合作博弈参与者, 每一个节点都观察网络中局势的变化, 寻求使得自身收益(Shapley 值)最大的节点或簇合作。为保证公平性每一轮决策中, 每个节点只有一次机会作出选择。每一个节点作出决策后得到的局势称为策略环境, 节点通过多轮决策, 当所有节点在某个策略环境下获得 Shapley 值都不再增大时停止初检测。

初检测算法具体步骤如下:

- 1) 初始数据为网络 $G = (V, E)$ 的拓扑结构, 即邻接矩阵 A 。
- 2) 定义动态集合 C_{result} , 初始化为 N 个独立节点联盟; 初始化一个集合 S_0 , 其元素为网络中的 n 个节点, 且固定不变。 C_{result} 称为当前策略环境。
- 3) 设置标识布尔型数组 $Flag[N]$, 对应每一个节点。标识每一个节点在当前策略环境下合作, 获得的 SH 值是否增大了。
- 4) 置每个 $Flag[i] = false$, 默认值全都为 false。
- 5) 依次循环 S_0 当中的每一个节点, 与当前集合 C_{result} 中的不包含自身的其他联盟合作, 根据 $SH_v(S, i)$ 公式计算获得的 SH 值, 若能获得 SH 值大于在原来联盟中的 SH 值, 则选择退出原来联盟加入使其获得更大 SH 值的联盟合作, 并设置对应节点 $Flag[i]$ 为 True; 若不能, 则不做任何操作。
- 6) 判断 S_0 中节点是否循环结束, 若结束求表达式 $Flag[0] \text{ or } Flag[1] \text{ or } \dots \text{ or } Flag[N]$ 的值, 如果表达式为 True(表示存在任意一个或多个节点的 SH 值增大了), 跳转执行步骤 4、5, 否则退出算法, 返回初步检测结果 C_{result} (表示所有节点的 SH 值都没有增大, 达到 SH 平衡)。

3.3. 社区调整算法

根据初检测算法, 会得到一些不合理、无意的小簇, 这些簇不具有社区的典型特征——社区内部连接紧密, 社区之间连接稀疏(一般社区内部边 > 社区外部边), 即不可以称之为社区。为了解决该问题, 使得最终社区检测结果中的簇具有明显的社区结构, 设计了调整这些小簇的社区调整算法。通过调整最终将网络划分成了多个具有明显社区特征的社区。社区调整算法步骤如下:

- 1) 计算簇 C_i 的内部边为 P_{in} , 外部边为 P_{out} 。
- 2) 若 $P_{out} \geq P_{in}$, 计算社区 C_i 与其他各簇之间的连接边数, 找出与 C_i 连接边数最大的簇 C_j 。
- 3) 将 C_i 与 C_j 进行合并。

该社区调整方案的合理性可以基于合作博弈的群体理性得到解释。假如簇 C_i 与 C_j 进行合并, 由公式(2)可得, 簇 C_i 中每个节点 x 的 SH 值增量为 $\Delta SH_x = \frac{1}{2} \sum_{j \in C_j} \left(\frac{a_{ij}}{d(i)} + \frac{a_{ji}}{d(j)} \right)$, 由此得簇 C_j 与 C_i 合并时,

社区 C_i 所有节点的 SH 值增量的总和为 $\Delta SH(C_i) = \frac{1}{2} \sum_{i \in C_i} \sum_{j \in C_j} \left(\frac{a_{ij}}{d(i)} + \frac{a_{ji}}{d(j)} \right)$ ，若要使得 $\Delta SH(C_i)$ 最大，可以近似的看成与 C_i 连接边最多的簇 C_j ，可以使得 $\Delta SH(C_i)$ 最大。

3.4. 基于合作博弈的社区检测算法

基于合作博弈的社区检测算法(Community Detection based on Cooperative Game, 简称为 CDCG)分为初检测和社区调整两个步骤，在本章 3.2 和 3.3 节已经分别详细阐述了两个算法的步骤，本节给出算法 CDCG 伪代码描述。

算法 CDCG 伪代码描述:

输入: $N = \{1, 2, \dots, n\}$, //网络中节点的集合
 $A = (a_{ij})_{n \times n}$, $i, j \in N$, //网络的邻接矩阵
 输出: C_{result}

- (1) $Flag[n] = false$;
- (2) $S = \{\emptyset\}$; //定义一个集合 S
- (3) *for* $i = 1$ *to* n
- (4) $S_i = \{i\}; S = S \cup \{S_i\}$;
- (5) *endfor*
- (6) $C_{result} = Initiate(A, N)$;

//初始化联盟，将一个节点作为一个联盟

- (7) *While* ($Flag[1]$ *or* $Flag[2]$ *or* \dots *or* $Flag[n]$)

//仍然存在节点在其他联盟中获得较大值

- (7.1) $Flag[1 \dots n] = false$;
- (7.2) *For each* $S_i \in S$

//测试每一个节点加入各社区,获得最大 SH 值则加入

- (7.2.1) *If* $Max(SH_v(C_r \cup S_i, x)) > SH_v(C_k, x)$

where $C_r \in C_{result}, x \in S_i$

- (7.2.2) *then* $\{C_r = C_r \cup S_i; Flag[i] = true\}$;

//如果节点 SH 值增大，给标识赋值 true

If $S_i \in C_k, then C_k = C_k - S_i$

//或者直接将该点从 C_k 中去除。

endfor //end 7.2

endWhile /*end 7

- (8) $C_{result} = Adjust(C_{result})$; //社区调整
- (9) *Output*: C_{result} //输出最终社区划分

时间复杂度分析:

若在网络中有 n 个节点，在算法中 *While* 循环是算法主要的时间耗费，步骤 7.1 时间复杂性为 n ；步骤 7.2， S 中元素个数为 n ； C_{result} 中最坏情况也 n 个元素；因此时间复杂性最坏情况为 n^2 。若执行 k 轮决策，可以使得每一个节点的 SH 值不再增大，则步骤 7 的时间复杂度约为 $O(kn^2)$ 。步骤 8 的时间复杂度最坏情况为 $O(n^2)$ 。因此，本文合作博弈社区检测算法时间复杂度为 $O(kn^2)$ ，其中 $0 < k < n$ 。

3.5. 优化剪枝策略

CDCG 算法主要时间耗费在初检测过程, 但该过程存在冗余计算与比较。因此, 为了更好的提高算法的时间效率, 根据初检测算法执行策略, 定义了无贡献节点和已归属节点。无贡献节点是指一个节点与其合作的对象(节点或簇)无任何连接边。该节点不会对其自身及合作对象产生 SH 值贡献, 可以进行剪枝。已归属节点是指一个节点的度已经全部存在于一个簇中时, 它 SH 值已经不会再增大。因此, 该节点已经不需要参加下一轮决策, 可以进行剪枝。

在 3.5.1 节与 3.5.2 节分别给出了无贡献节点和已归属节点的形式化定义, 给出了无贡献节点剪枝与已归属节点剪枝的伪代码描述。

3.5.1. 无贡献节点剪枝

定义 2 无贡献节点

网络邻接矩阵 $A = (a_{ij})_{n \times n}$, $i, j \in N$, 存在节点 i 和当前联盟 $C_x = \{j, k, m, \dots\}$, 若任意的 $a_{ij}, a_{ik}, a_{im}, \dots$ 均为 0, 则称 i 为联盟 C_x 的无贡献节点。

无贡献节点剪枝方法伪代码描述:

```
bool Test_Contribution(v, C, A) { /* v 为当前节点, C 为当前联盟, A 为邻接矩阵 */
    Flag = 0;
    for i = 1 to C.Length // 判断是否有边相连
        if A[v, C[i]] = 1 then
            Flag = 1;
            break;
        endif
    endfor
    if Flag = 1 { return true; }
    else { return false; }
}
```

3.5.2. 已归属节点剪枝

定义 3 已归属节点

网络邻接矩阵 $A = (a_{ij})_{n \times n}$, $i, j \in N$, 存在节点 i 属于联盟 $C_x = \{i, j, k, m, \dots\}$, 设节点 i 在整个网络中的度为 $d(i)$, 且节点 i 在 C_x 内的度为 $d_x(i)$, 若 $d(i) = d_x(i)$, 则称节点 i 为已归属节点。

证明: 存在网络 $G = (V, E)$, 其邻接矩阵为 $A = (a_{ij})_{n \times n}$, $i, j \in N$, $d(i)$ 表示节点 i 在网络中的度。若有节点 i 属于联盟 $C_x = \{i, k, m, z, \dots\}$, $d_x(i)$ 表示节点 i 在 C_x 内的度。

$$\because d(i) = d_x(i)$$

$$\Rightarrow \sum_{j=0}^n a_{ij} = a_{ik} + a_{im} + a_{iz} + \dots$$

\Rightarrow 节点 i 仅与 C_x 内的节点有连接, 则当有任意节点 p 加入当前联盟 C_x ($a_{ip} = a_{pi} = 0$), 根据

$$SH_v(C_x, i) = SH_v(C_x, i) + \frac{1}{2} \left(\frac{a_{ip}}{d(i)} + \frac{a_{pi}}{d(p)} \right) = SH_v(C_x, i)$$

\Rightarrow 节点 i 的 SH 值不再变化且达到最大。

\Rightarrow 节点 i 为已归属节点。

已归属节点剪枝方法伪代码描述:

```

bool Test_belong(v,C,A){
//v为当前节点, C为联盟, A为邻接矩阵
d_x(v)=0; d(v)=\sum_{j=1}^n A[v,j];
for i=1 to C.Lenght
if A[v,C[i]]=1
d_x(v)++;
endif
endfor
if d_x(v)=d(v) then {return true;}
else {return false;}
}

```

4. 实验分析

本文将从算法社区检测的效果、时间效率等方面对本文所提算法进行评估。算法采用的编程语言为C#, 开发平台: Microsoft Visual Studio 2010; 实验使用的计算机配置: Intel(R)Core(TM) i3-2330M CPU @ 2.20 GHz, 4G 内存; 64 位操作系统: Windows7 操作系统;

4.1. 实验数据

1) 128 个节点的 ad hoc 网络。该网络在多篇文章中作为典型基准网络[2] [10]-[12], 用于验证社区检测算法的有效性。它主要特点是已知社团结构, 具体构造如下: 选取 $n = 128$ 个节点, 分成 4 个社团, 每个社团包含 32 个节点, 网络的平均度为 16, Z_{out} 为某节点与属于其它社团节点之间连接的期望值, Z_{out} 越大社区结构越模糊, 通过控制参数 Z_{out} , 观察社区划分情况。

社区检测的准确性用归一化互信息(Normalized Mutual Information, NMI)来衡量[11]。归一化互信息定义如下:

$$NMI(X, Y) = \frac{-2 \sum_{i=1}^{C_x} \sum_{j=1}^{C_y} M_{ij} \log \left(\frac{n M_{ij}}{M_{i \cdot} M_{\cdot j}} \right)}{\sum_{i=1}^{C_x} M_i \log \left(\frac{M_{i \cdot}}{n} \right) + \sum_{j=1}^{C_y} M_j \log \left(\frac{M_{\cdot j}}{n} \right)}$$

X 为实际的社区划分, Y 为算法得出的社区划分, $NMI(X, Y)$ 表示算法得出的社区划分结果与实际社区划分之间的相似程度, 取值范围为 0~1, 其值越大算法所得的社区结构与真实社区结构越接近, $NMI = 1$ 表示所得结果与真实社区划分完全一致。 $NMI = 0$ 时表示算法得到的划分与真实划分完全不同。公式中的 M 为一个模糊矩阵, 其中行对应真实社区, 列对应算法找到的社区。 M 中的一个元素 M_{ij} 表示出现在找到的社团 C_i 中的真实社团 C_k 中的节点数目。 C_x 为真实社区的数目, C_y 为算法找到的社区的数目。

2) 空手道俱乐部网络。这个网络是由 Zachary [13] 在观察一所美国大学空手道俱乐部成员之间的社会联系而构建的。由于俱乐部的管理者和主教练之间发生争吵, 于是俱乐部分裂成两个小俱乐部。在空手道俱乐部网络中有 34 个节点, 每个节点表示俱乐部中的一个成员。在文献[13]中, Zachary 给出了该网络自然划分如图 3 所示, 白色和黑色分别代表不同的社区。这个网络广泛地应用于验证网络社区检测的算法。

3) **海豚网络**。海豚网络是由 Lusseau [14]对 62 只尖嘴海豚之间的频繁联系所构成的无向社会网络。这个网络有 62 个节点，被广泛认同的 GN-2 社区划分如图 8 所示。该网络也广泛地应用于验证网络社区检测的算法。

4) **LFR 基准网络**[15]。LFR 基准网络是一种构造的现实主义的网络，能够用于研究社区结构，被广泛应用于测试社区检测算法的性能。LFR 基准网络的特点是节点度和社团规模的非均匀性，它能根据用户指定的分布生成更接近真实情况的网络。这里分布不仅包括网络中节点度的分布，而且包括各个社区中节点数的分布。文献[15]给出了 LFR 基准网络构造算法，源程序来源于 Santo Fortunat 的网站：

<https://sites.google.com/site/santofortunato/inthepress2>。

LFR 基准网络：LRF(N, k, maxk, mu, minc, maxc)，其中参数 N 为节点个数，k 为平均度，maxk 为最大度，mu 为节点外部/总度数，minc 为最小社区中节点个数，maxc 为最大社区中节点个数。调整参数 N 变化网络的规模，固定其他各参数变量得到不同规模的网络。本文实验中将利用该模型构建不同规模的网络对比算法的时间效率。

4.2. 实验分析

1) **128 个节点的 ad hoc 网络**。通过控制 Z_{out} ，从 1 变化到 8，比较了 GN 算与本文算法 NMI 值的变化，如图 1 所示。由于该网络生成具有随机性，对于每个 Z_{out} 本文重复做了 100 次实验，各数值是 100 次重复试验的平均值。由图可以看出当 Z_{out} 从 1 增加 8 时，该模拟网络社团结构由清晰变得模糊，所以 NMI 曲线呈现递减趋势。本算法在 $Z_{out} = 1\sim 4$ 正确划分准确率为 100%， $Z_{out} \geq 5$ 之后，准确性开始下降，随着 Z_{out} 值的增大，其下降的速度要快于 GN 算法。GN 算法 $Z_{out} \geq 6$ 之后，准确性才开始下降。因此本文算法在社区结构比较明显的网络中，检测效果较好。

2) **空手道俱乐部网络**。运用该网络比较了本文算法初检测的社区效果与社区调整后的社区效果，比较了 GN 算法与本文算法的社区检测的结果。从图 2 与图 4 看出，初检测结果出现了规模小的、不合理的社区，在社区调整后得到了较为合理的社区划分。观察比较图 3 与图 4，本文算法在空手道俱乐部网络中划分效果与自然划分有所区别，将 {5,6,7,11,17} 作为了一个单独的社区划分，节点 10 划分错误，其他各节点的划分均与实际社区划分一致。与 GN 算法比较，本文算法划分成了 3 个社区，GN 算法划分成了 2 个社区，如图 5 所示。本文算法划分成 3 个社区，具有合理性。若从模块度(Q 值)的角度网络自然划分时 Q 值为：0.365；GN 算法划分最大 Q 值为：0.353；本文算法划分结果的 Q 值为：0.396。从以上

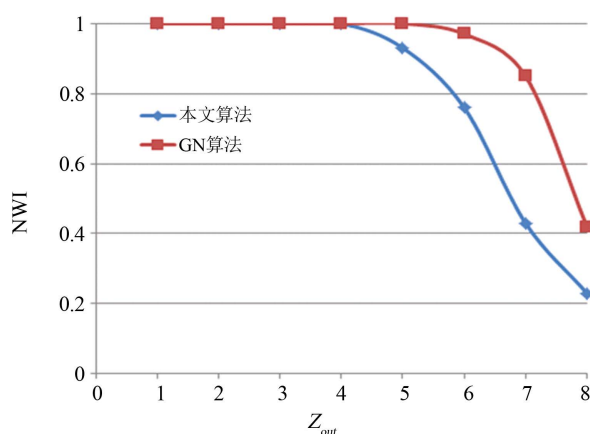


Figure 1. Comparison of community detection accuracy

图 1. 社区检测准确度比较

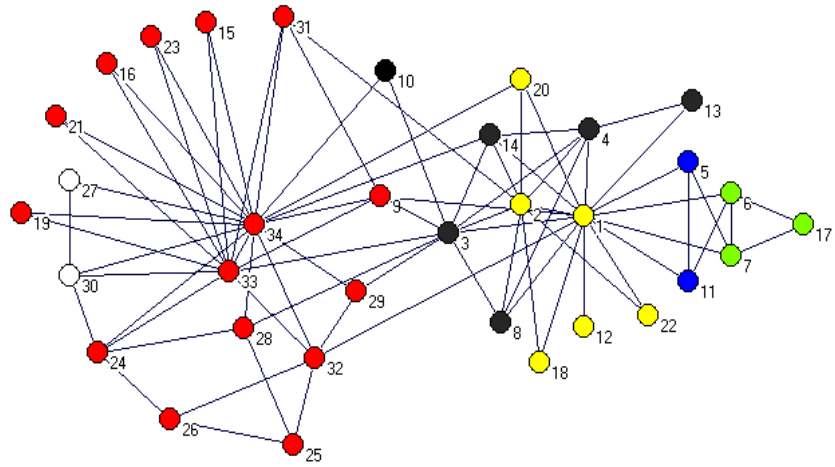


Figure 2. Effect of initial detection of the community

图 2. 初检测的社区效果

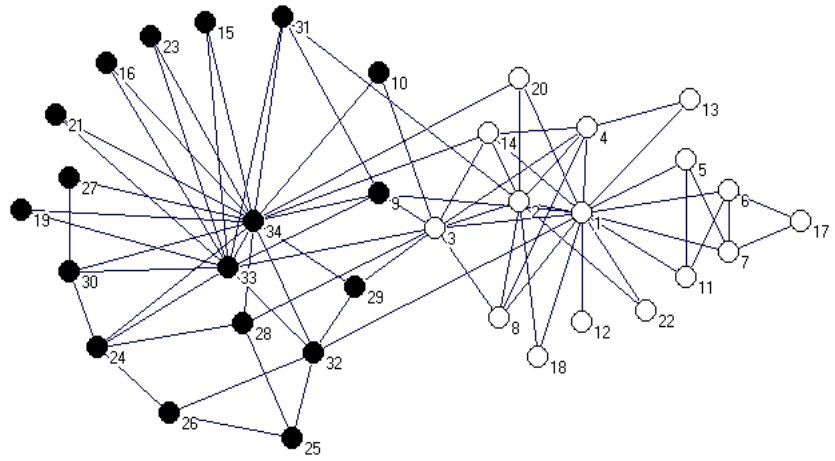


Figure 3. Natural division of the Karate Club network [13]

图 3. 空手道俱乐部网络的自然划分[13]

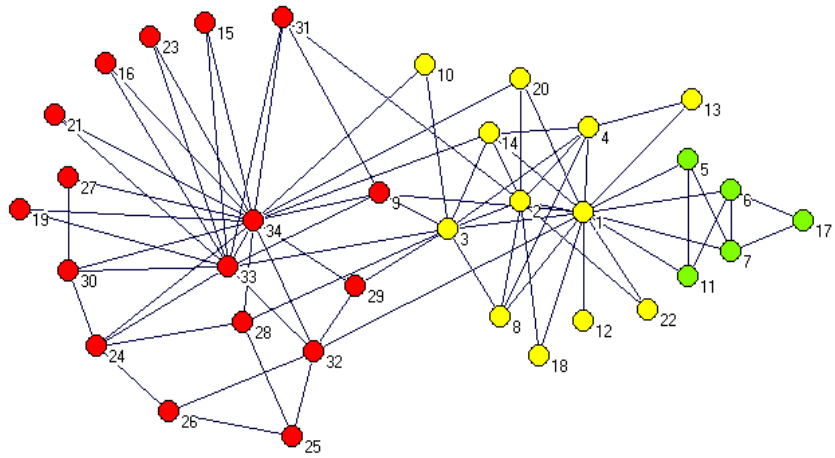


Figure 4. The effect of community adjusted community

图 4. 社区调整后社区效果

分析知，本文在俱乐部网络中社区划分效果良好，且算法结束得到确定的社区个数。

3) **海豚网络**。运用该网络比较了本文算法在初检测的社区效果与社区调整后的社区效果；比较了本文算法与 GN 算法-2 社区状态下的社区检测效果。观察图 6 与图 7，可知在初检测结果中由于一些边缘节点及形成闭合回路的节点，造成了一些小的簇出现，在社区调整后，很好的将这些簇进行了归属。从图 7 和图 8 对比知，仅有节点 8，节点 20 与 GN 算法-2 社区情况下划分不同，说明本文算法在海豚网络中社区划分效果良好。

4) LRF 随机基准网络。

图 9 利用 LRF 基准网络，随机生成了网络中节点个数分别为 100,200,300,400,500,600 的数据集，比较了 GN 算法、与本文算法的时间效率。随机生成 LRF 基准网络采用参数列表如下：

$$\text{LRF}(N, k = 8, \text{maxk} = 15, \mu = 0.1, \text{minc} = 10, \text{maxc} = 30);$$

保持其他参数不变，改变网络规模 N ，随机生成不同规模的 LRF 基准网络。由于 GN 算法与本文算法在网络中节点个数相同的情况下运行时间差异较大，为更好的比较二者之间的时间效率，将实际运行时间取以 10 为底的对数进行处理，该函数为单调递增函数，不会使曲线趋势发生变化，且直观的反映数

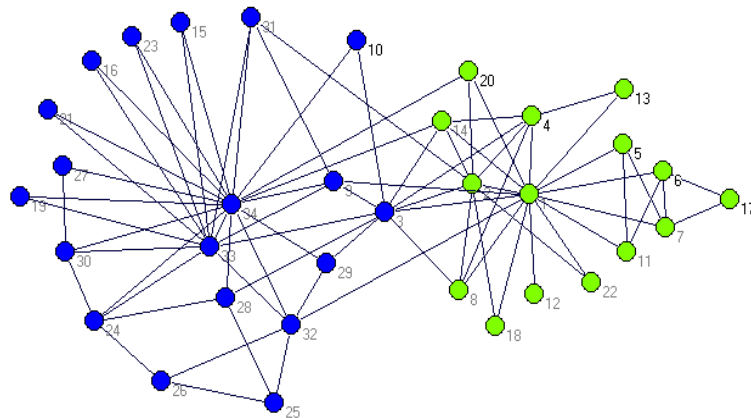


Figure 5. GN algorithm-maximum Modularity of community detection effect [2]
图 5. GN 算法 - 最大模块度时的社区效果[2]

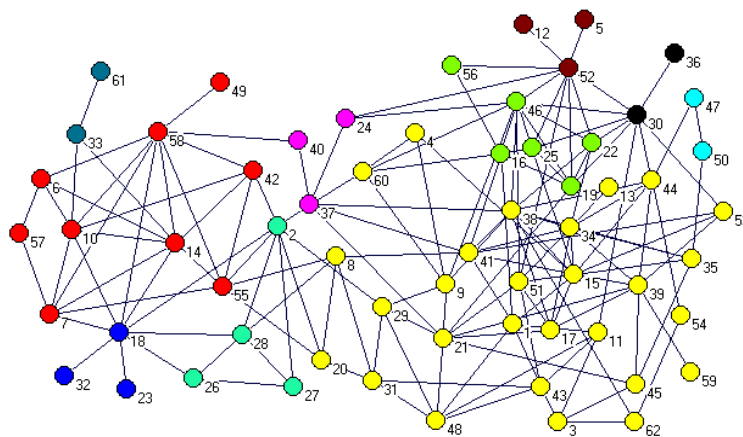


Figure 6. Effect of initial detection of the community
图 6. 初检测的社区效果

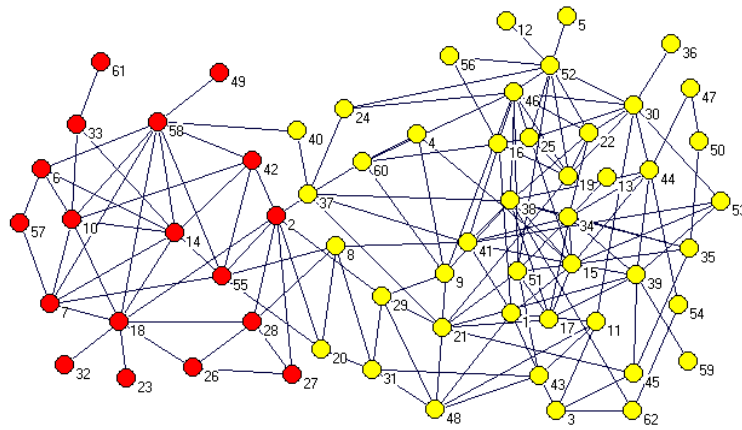


Figure 7. The effect of community adjusted community
图 7. 社区调整后社区效果

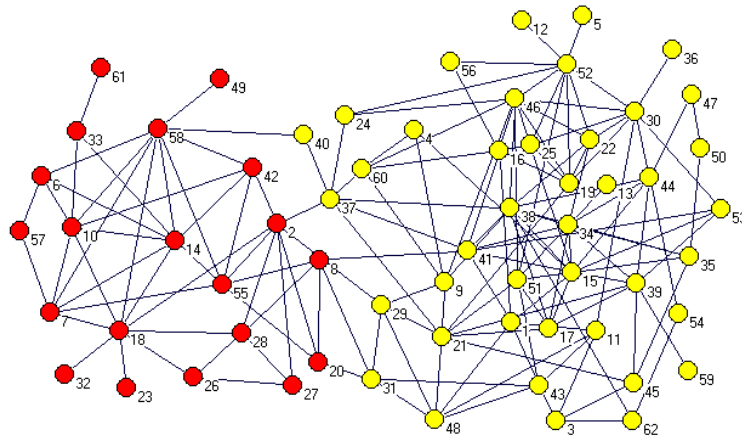


Figure 8. GN algorithm-effect of The Dolphin network is divided two community
图 8. GN 算法 - 海豚网划分为 2 个社区的效果

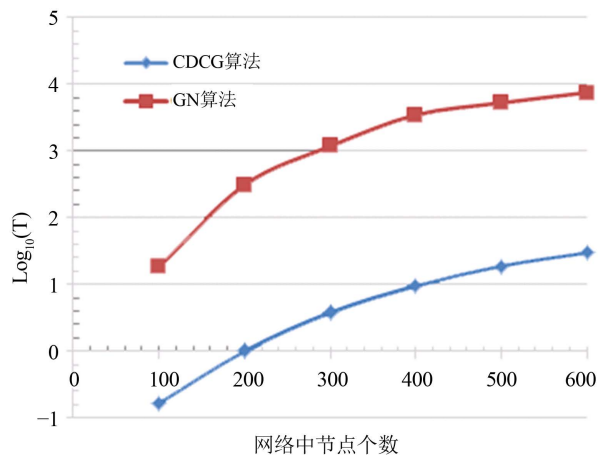


Figure 9. Efficiency comparison of GN algorithm and CDCG algorithm
图 9. GN 算法与本文算法时间效率比较

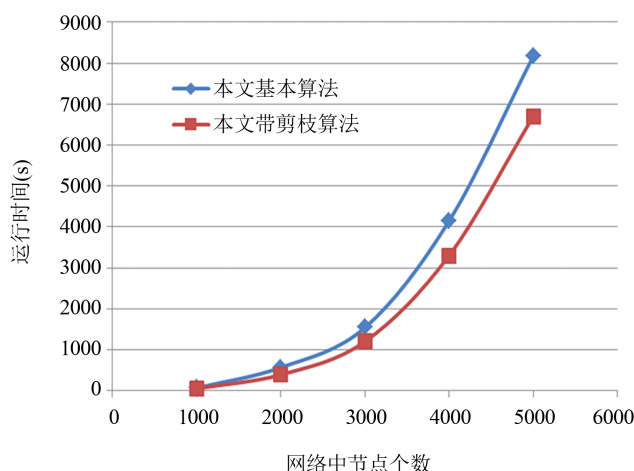


Figure 10. Time effect of pruning CDCG algorithm
图 10. 带剪枝算法运行时间效果

量级区别。图 9 中横坐标为网络规模，纵坐标为 $\log_{10}(T)$ ， T 为算法实际运行时间。从图 9 可以看到，GN 算法、本文算法随着网络中节点数目的不断增大，算法运行时间呈上升趋势，但本文算法运行时间明显优于 GN 算法且在相同网络规模下是数量级的差距。说明随着网络规模的增大有更好的时间效率。

图 10 利用 LRF 基准网络，随机生成了网络中节点个数分别为 1000, 2000, 3000, 4000, 5000 的数据集，比较了不带剪枝合作博弈检测算法与带剪枝算法时间的效率。LRF 基准网络采用参数列表如下：

$$\text{LRF}(N, k = 15, \text{maxk} = 50, \mu = 0.1, \text{minc} = 20, \text{maxc} = 50);$$

由图 10 可以看出，随着网络规模的增大，算法的运行时间呈上升趋势。带剪枝的算法具有一定的剪枝效果。但节点规模较小时，剪枝效果不明显。当节点的平均度不发生变化时，增大网络的规模，无贡献节点就会增多，且随着算法的执行有更多的节点很快找到联盟归属，因此，带剪枝的算法会在节点平均度不变时，网络规模越大，剪枝效果越明显。但如果网络规模的增大，节点平均度数增加，可能将不会有明显剪枝效果。

5. 总结

本文探索了一种在社会个体行为基础上的合作博弈的社区检测算法，该算法由初检测和社区调整两部分组成。首先，介绍了文中采用的合作博弈的特征函数及高效计算 Shapley 值公式，其次，根据合作博弈中个体理性为核心设计了初检测算法，在初检测算法结果的基础上，利用社区内部连接紧密，社区之间连接稀疏的特征，判断出不合理、无意的小簇，设计调整算法。为了更好的提高算法效率，提出了无贡献节点剪枝策略及归属节点剪枝策略。最后，编程实现了基于合作博弈的社区检测算法，通过实验分析验证了本文算法具有较好社区检测效果及时间效率。

基金项目

国家自然科学基金(61262069)。

参考文献 (References)

- [1] 孟小峰, 余力 (2011) 用社会化方法计算社会. *中国计算机学会通讯*, 7, 25-30.
- [2] Zhou, L.H., Cheng, C., Lv, K. and Chen, H.M. (2013) Using coalitional games to detect communities in social net-

- works. *Web-Age Information Management*, Springer, Berlin, 326-331.
- [3] Girvan, M. and Newman, M.E.J. (2002) Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, **99**, 7821.
 - [4] Zhao, Z.Y., Feng, S.Z., Wang, Q., Huang, Z.X., Williams, G.J. and Fan, J.P. (2012) Topic oriented community detection through social objects and link analysis in social networks. *Knowledge-Based Systems*, **26**, 164-173.
 - [5] Shi, C., Yan, Z.Y., Cai, Y.N. and Wu, B. (2012) Multi-objective community detection in complex networks. *Applied Soft Computing*, 850-859.
 - [6] Tasgin, M., Herdagdelen, A. and Bingol, H. (2007) Community detection in complex networks using genetic algorithms.
 - [7] Pizzuti, C. (2009) A multi-objective genetic algorithm for community detection in networks. *The 21st IEEE International Conference on Tools with Artificial Intelligence*.
 - [8] Ahn, Y.-Y., Bagrow, J.P. and Lehmann, S. (2010) Link communities reveal multiscale complexity in networks. *Nature*, **466**, Article ID: 09182.
 - [9] Chen, W., Liu, Z.M., Sun, X.R. and Wang, Y.J. (2011) Community detection in social networks through community formation games. *Proceedings of IJCAI*, **3**, 2576-2581.
 - [10] Fortunato, S. (2010) Community detection in graphs. *Physics Reports*, **486**, 75-174.
 - [11] Danon, L., Diaz-Guilera, A., Duch, J. and Arenas, A. (2005) Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, Article ID: P09008.
 - [12] Newman, M.E.J. and Girvan, M. (2004) Finding and evaluating community structure in networks. *Physical Review E*, **69**, Article ID: 026113.
 - [13] Zachary, W.W. (1977) An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, **33**, 452-473.
 - [14] Lusseau, D. and Newman, M.E.J. (2004) Identifying the role that animals play in their social networks. *Proceedings of the Royal Society*, **271**, 477-481.
 - [15] Lancichinetti, A., Fortunato, S. and Radicchi, F. (2008) Benchmark graphs for testing community detection algorithms. *Physical Review E*, **78**, Article ID: 046110.