山西大学

2016 届硕士学位论文

大规模复杂网络的社区发现算法研究

作者姓名　　梁　晋

指导教师　　梁吉业　教 授

学科专业　　计算机应用技术

研究方向　　社会网络分析

培养单位　　计算机与信息技术学院

学习年限　　2013 年 9 月－2016 年 6 月

二〇一六年六月

Thesis for Master’s degree, Shanxi University, 2016

Research on Community Detection Algorithms for Large Complex Network

Student Name	Liang Jin
Supervisor	Prof. Liang Jiye
Major	Technology of Computer Application
Specialty	Social Network Analysis
Department	School of Computer and Information technology
Research Duration	Sep.2013——Jun.2016

June 2016

目 录

Contents

中 文 摘 要

针对社区发现算法的研究已经成为社会学、计算机科学、生态学和经济学等许多领域研究中最重要的课题之一。随着近年来互联网高速发展和移动终端的普及应用，使得复杂网络的种类和规模得到了快速的发展和变化，许多网络中节点的数目已达到数亿级并仍保持着快速增长趋势。上述这些现状给社区发现算法的研究带来了许多新的困难。

本文的主要研究内容如下：

（1）详细介绍主流社区发现算法的原理以及相关步骤，并分类介绍了目前适用于较大规模网络的社区发现算法，并在时间效率、准确度、应用范围和优劣势等方面进行分析比较。

（2）提出了一种基于抽样的大规模网络社区发现算法，该算法首先利用基于度的随机游走技术对整体网络进行抽样得到子图，然后将基于概要的发现算法应用到此子图上进行社区发现，之后获得初始社区，最后依据已有初始社区与未抽样的节点的相似度迭代式地将剩余节点进行划分。通过与另外的算法进行比较分析，表明了该算法能够在保证准确性的同时提高计算效率。

（3）提出了一种基于压缩的大规模网络社区发现算法，根据一些复杂网络的长尾特性对复杂网络进行折叠压缩，选取核心节点之后完成整个网络的社区划分。通过与另外的算法进行比较分析，表明了该算法能够在保证准确性的同时提高计算效率。

本文的研究工作尤其是提出的两种算法能够在不丢失社区发现质量的同时很大程度上缩短算法的运行时间，使此项研究能够很好的适应复杂网络规模快速增大的现状，具有重要意义。

关键词：大规模复杂网络；社区发现；抽样；压缩；模块度

ABSTRACT

With the rapid development of Web 2.0 and the rise of online social networks, computational efficiency of community detection in large-scale social networks has become a major problem. Although research achievements are numerous at present such as the Louvain method, Clustering-based method etc, most of these achievements cannot be adopted in large-scale social networks because of the heavy computation and the decrease of accuracy.

The main content is as follows:

(1) We introduced some mainstream community detection algorithms and some algorithms which were suitable for the large-scale complex networks in time and accuracy.

(2) We proposed a community detection algorithm based on sampling and label propagation. In particular, the proposed algorithm firstly generated some subgraphs via random walk sampling and computed the weight of each edge, then detected communities on these subgraphs. In the end, we partitioned the unsampled nodes into the communities according to subgraphs' structures. The experiments were conducted in comparison with widely used state-of-the art community detection algorithms on several real networks. The results showed that the proposed algorithm can provide computational efficiency, while maintaining the effectiveness.

(3) We proposed an algorithm based on folding and label propagation. The proposed algorithm folded the large-scale complex networks according to long tail characteristics of complex networks, and then chose some core nodes on folded networks. In the end, we partitioned remain nodes into the communities. After experimental comparison, the results showed that the algorithm can efficiently and effectively find the community structure.

The proposed two algorithms can provide computational efficiency, while maintaining the effectiveness of community detection, which make the

community detection algorithm study well adapt to the current situation of the rapid increasing scale of complex network.

Key words： Large-scale complex network； Community detection； Sampling；Folding；Modularity

第一章 绪论

1.1 研究背景与意义

1.1.1 研究背景

从上个世纪九十年代以来，针对复杂网络的研究已经成为许多领域研究中最重要的课题之一，比如社会学、计算机科学、生态学和经济学等诸多领域。细胞是一个由多种化学物质通过化学反应连接的复杂网络、Internet 是电脑和路由器间通过各种物理或无线链接形成的复杂网络、潮流和思想的传播促成了各种社会关系的形成，构成了人类的社交网络、万维网是由很多网页通过超链接的方式连接起来进而形成的一种庞大且虚拟的复杂网络等。

近年来随着计算机革命进程的加快、互联网高速发展和移动终端的普及应用，使得复杂网络的种类和规模得到了快速的发展和变化，许多网络中节点的数目已达到数亿级并仍保持着快速增长趋势。例如：作为中国最知名的社交网站之一，微博在截至 2015 年 9 月底时，其月活跃用户数已经达到 2.22 亿人，比上年同期增长 33%；2015 年 8 月底，社交网络服务网站 Facebook 在单日已有突破十亿的活跃用户。这些现状表明人们之间的交流已经不再只是通过面对面的方式，而是对于社交网络的线上交流愈加倚重。

随着相关研究的不断深入，科研人员发现复杂网络呈现出三个普遍特性：小世界性，节点度数的幂律分布性质和社区结构性质。

小世界性是指，虽然复杂网络通常包含的节点和边的个数较多，但是其中任意两个节点间的路径往往是比较短的。最著名的小世界性表现是由社会心理学家 Stanley 于 1967 年提出的六度分离概念，他通过一系列的实验，尝试证明完全陌生的两个美国人之间至多只需要五个中间节点就可以建立起联系。小世界性可以描述和解释一些复杂网络中的某些表现，已在很多领域得到了应用，如保险和直销行业，用以发掘潜在客户。

节点度数的幂律分布性质是指网络中节点的度数是呈幂律分布，具体表现是网络中大部分的节点的度数都很小。现实中的多数网络都符合这一特性，如万维网和金融网络等。

社区结构性质具体表现是，网络通常可以按照某种规律划分成很多不同的社区，各个社区的职责和拥有的成员不同，处于同一社区的成员之间通常具有相同的兴趣或是联系比较紧密，而不同社区的成员联系较为稀少。针对社会网络的社区研究自

二十世纪六十年代开始已经得到了广泛的关注。不止是在社交网络，社区结构同样出现在各个类型的真实网络中。在蛋白质交互网络中，处于不同社区的蛋白质功能往往不相同[1]；在万维网中，处于相同社区的页面具有相近或相同的主题[2]；在生物的代谢网络中，相同社区表示拥有着相同的功能[3]。现如今互联网的迅速扩展使得出现了基于互联网交流的在线虚拟社区[4]。

1.1.2 研究意义

对社区结构的发现研究有以下意义：

第一，在应用方面的意义。发现各种复杂网络中的社区结构有着很广泛的应用前景，进而对人们的生活有着很大的积极影响。对万维网上的客户进行社区发现，结果表明处于同一社区的客户往往都有相似的兴趣[5]，基于这一特性，以知名购物网站之一的亚马逊网站为例，可以根据某一客户的兴趣为某一在线零售商寻找处于同一社区的潜在客户，进而建立高效的推荐系统，更好地指导零售商进行项目列表的改进并提高商业机会。对一些大型图表如气象图、地图等进行社区发现之后，用以创建数据结构能够有效地存储相关的图形数据并提供高效的导航查询，如路径搜索[6]。自配置网络如 Ad hoc（特设网络）是由位于同一地区的代理节点进行通信所形成的并且在不断变化的网络，该网络本身不包括集中维护路由表，对该网络进行社区发现能够帮助生成一个紧凑路由表使得之前的通信路径仍然有效[7]。在计算科学的并行计算中，通过社区发现可以获得较好的方式来分配任务处理器，减少处理器之间的通信，进而达到快速计算的目的。

第二，社区发现对很多学科的进一步发展有着直接或间接的影响，在科学研究中具有重要意义。同一社区中不同节点在社区中的作用是不相同的。在社区中心的节点通常与处于同一社区的节点有着大量的连边，这类节点往往控制和维护着社区的稳定，而处于社区边缘的节点发挥的是与其它社区进行交流的媒介作用，社区中不同节点的不同作用对社会网络研究和新陈代谢网络研究具有比较突出的积极意义[8] [9]。与此同时，社区发现实际上是完成了对复杂网络的粗粒化过程，对原始网络重新进行了描述，揭示了社区之间的关系并将网络的规模进行了压缩，能够帮助解决如今大数据时代下各个学科处理数据难的瓶颈问题，使得计算效率得到大大提高。

第三，社区发现能够揭示真实网络中的一些潜在规律，帮助人们更好地了解世界。真实世界中的网络社区都是层次化的。以公司为例，公司是由不同的部门构成的，同时每个部门下又可分为不同的小组。诸如此类的例子比比皆是，复杂网络中的层次结构及其演化对揭示世界中复杂系统的规律发挥着关键作用[10]。一个复杂网

络（系统）的更新或进化速度远远小于组成其的各种社区（系统）的更新进化速度，因为社区（系统）达到稳定状态要比整个复杂网络（系统）达到稳定更为容易一些，同时整个复杂网络（系统）完成更新进化的前提是其各个社区已经达到稳定状态。人们可以基于真实网络的此种特性，利用社区发现深入挖掘复杂网络下隐藏的一些规律并对整体网络的未来形态进行准确预测，为人类的进步做出一定贡献。

1.2 国内外研究现状

1.2.1 社区发现研究现状

社区发现算法顾名思义就是从复杂网络中发现社区结构。针对社区发现算法的研究已经成为社会学、计算机科学、生态学和经济学等许多领域研究中最重要的课题之一。

在社会网络研究中，针对社区发现的研究是由 Weiss 和 Jacobson 为一个政府机构识别工作小组而率先展开的[11]，他们首先利用私人采访的方式了解该工作机构的员工之间的关系，并对此建立了关系矩阵，再通过不断移除不同小组间的连边进而完成工作小组的识别工作，他们提出的去除“桥梁”的想法是现代社区发现算法的基础。Stuart 通过基于相似性的投票模式在政治机构中寻找相同政治信仰的社区。George 通过对关系矩阵的行和列进行重新排列形成广义分块对角矩阵进而完成社区结构的划分，此方法现已成为社区发现算法的一个标准步骤[12]。与此同时，在社会网络中传统社区发现算法可以分为层次聚类方法和分割方法，它们都是基于社会网络中个体之间的相似度实现社区划分的[13]。

2002 年 Girvan 和 Newman 发表了社区发现领域里的开创性论文，他们提出了一种新的算法迭代式地计算边的介数并将介数最大的边删除直至将复杂网络划分成各个社区[14]。其中介数是指通过该边的最短路径的次数，是一种中心性度量指标，用以表示边所处角色的重要性。这篇论文对复杂网络研究有着很深远的意义，在此之后，社区发现的研究开始广泛利用计算机科学、非线性动力学、离散数学和社会学的许多概念与方法。总的来说，多学科不同技术和思想的交叉合作为社区发现算法的不断发展提供了保障。

在我国，来自清华大学、吉林大学、中科院等很多大学不同学科的老师和科研人员都对社区发现展开了细致且深入的研究，并取得了显著的成果，比如吉林大学的刘大有等人提出的快速社区发现算法[15]和结构相似度算法[16]为减少社区发现算法时间复杂度方面提供了新的思路，中科院计算研究所的程学旗等人提出的重叠社区

结构度量方法[17]为今后社交网络分析领域尤其是重叠社区结构的深入研究奠定了基础。

1.2.2 社区发现研究所面临的问题

几十年来国内外的研究人员针对复杂网络中社区结构已经投入了大量的研究，也取得了一些显著的成果，提出了一些经典算法。但是仍然存在着一些问题，值得科研人员去着重解决。这些问题主要体现在以下几个方面：

第一，重叠社区发现。例如：一个科研人员也可以同时在多个领域展开研究工作，而这些领域里的其他成员往往是不相同的；在生物网络中一个细胞在不同的组织中起着不同的作用。重叠性的社区在许多真实网络中具有普遍性[18] [19]。重叠社区发现有着广泛的应用前景，但是很多传统的发现算法所识别出的社区都是不相交的，不能适用于此情形。重叠社区的研究已经引起了各个学科科研人员的关注，同时也提出了一些重叠社区发现算法，但是通过对比不同算法的实验结果说明在重叠社区发现领域仍然有很大的改进空间[20]。

第二，动态社区发现。真实世界中的复杂网络大多是动态的，比如世界上最受欢迎的社交网络如 Facebook、Twitter 和 LinkedIn 都是在随着时间的推移而不断地进行着扩张或改变。用户的加入或退出引起会引起局部关系网络的变更，经过一段时间之后，这些局部变更会导致整体社会网络发生巨大的转变。虽然多次执行传统的社区发现算法可以完成对与动态社区发现相似的结果，但是存在一些不容忽视的问题：（1）昂贵的算法执行代价，尤其是当对象是大规模网络时；（2）会陷入到局部最优化的陷阱中；（3）针对只有局部变化的复杂网络得到的社区发现结果是一样的[21]。所以对动态社区发现进行有针对性的研究是极其复杂也是极其重要的问题，需要科研人员提出耗时少且又有效的社区发现算法。

第三，大数据时代下的社区发现。随着大数据时代的到来，产生了海量的数据，使得复杂网络中顶点和边逐渐增加，如在线社交网络和即时通讯网络等。现有的算法通常很难在如此海量的数据集上得以应用，因此需要通过改进技术手段或从不同的角度提出适用于大规模复杂网络的社区发现算法，使社区发现研究跟上大数据时代的脚步。

1.3 本文研究内容

本文的主要研究内容如下：

1、详细介绍主流社区发现算法，并对这些算法在时间效率、准确度和适用范围

等方面进行分析比较。

2、着重介绍了目前存在的适用于大规模网络的相关算法原理和代表算法。

3、提出一种基于抽样的大规模网络社区发现算法，通过与另外的算法进行比较分析，表明了该算法能够在保证准确性的同时提高计算效率。

4、提出一种基于压缩的大规模网络社区发现算法，通过与另外的算法进行比较分析，表明了该算法能够在保证准确性的同时提高计算效率。

1.4 本文的结构安排

本文的具体内容分为五章，各章结构和内容简介如下：

第一章 绪论。主要讲述本文相关领域的研究背景与意义。

第二章 社区发现算法综述。详细介绍了相关基础知识和主流算法的原理以及相关步骤，并分类介绍了目前适用于较大规模的算法，并在时间效率、准确度、应用范围和优劣势等方面进行分析比较。

第三章 提出了一种基于抽样的大规模网络社区发现算法，该算法首先利用基于度的随机游走技术对整体网络进行抽样得到子图，然后将基于概要的发现算法应用到此子图上进行社区发现，之后获得初始社区，最后依据已有初始社区与未抽样的节点的相似度迭代式地将剩余节点进行划分。同时通过实验与其它社区发现算法进行了对比分析。

第四章 提出了一种基于压缩的社区发现算法，根据一些复杂网络的长尾特性对复杂网络进行折叠压缩，选取核心节点之后完成整个网络的社区划分。同时通过实验与其它社区发现算法进行了对比分析。

第五章 总结与展望。

第二章 社区发现算法综述

2.1 基础知识

本小节主要介绍了一些与社区发现相关的基础知识，包括相关概念和数学定义公式。

2.1.1 图模型

复杂网络可以被抽象的看作是一个图，进而可以通过图论中的知识来进行数学表示。本文讨论的是无向无权网络，可以表示成网络$G(V,E)$，其中$V=\{V_i|i=1,...N\}$为网络中所有节点的集合，$E=\{e_{ij}|V_i,V_j\in V\}$为所有的边的集合，$e_{ij}$表示端点为$i$和$j$的一条边，$N=|V|$，$M=|E|$，即$N$为节点的总数量，$M$为边的总数量。图 2.1 为 Zachary 空手道俱乐部的网络图表示，其中$N=34$，$M=72$。

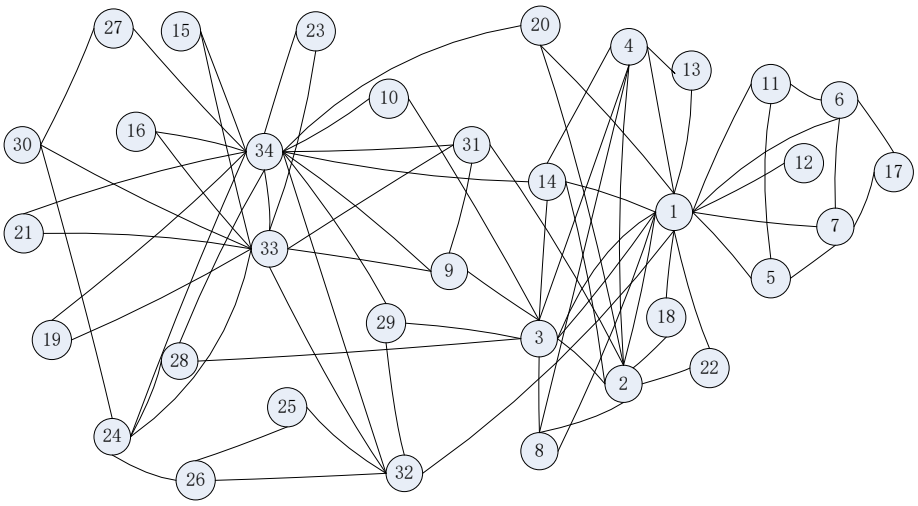


图 2.1 Zachary 网络图

2.1.2 度、平均度和度分布

节点的度是指该节点的邻居节点的数量，在图 2.1 中节点 6 的度为 4、节点 30 的度为 3。在本文中度用k_i表示节点i的度，给定网络$G(V,E)$的邻接矩阵$A=(a_{ij})_{N\times N}$，其具体数学计算公式为：

$$k_i=\sum_{j\in N}a_{ij} \tag{2.1}$$

平均度是指所有节点的度数平均值，记为$\bar{k}$，具体数学计算公式为：

$$\bar{k}=\frac{1}{N}\sum_{i,j=1}^{N}a_{ij} \tag{2.2}$$

度分布的分布函数用$P(k)$来表示，具体数学计算公式为：

$$P(k)=\frac{n(k)}{N} \tag{2.3}$$

均匀网络的度分布是如图 2.2 所示的泊松分布，其几何曲线形状以$\langle k\rangle$为分水岭，离$\langle k\rangle$越远其值越小。

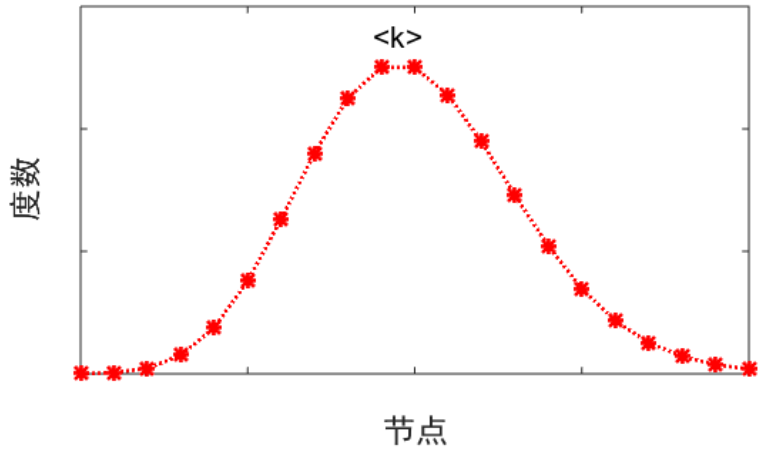


图 2.2 泊松分布

但随着近年来对真实世界网络研究的深入，人们发现其中大部分节点的度是比较小的，只存在极少数度值较大的节点。以 Facebook 朋友关系为例：只有少数的用户有着超过 2000 人的朋友数量，而大部分用户的朋友数量都不足 200 人。也就是说，这些网络的度分布并不同于图 2.2 所示，没有类似于$\langle k\rangle$的特征值，而是如图 2.3 有一个长长的尾巴，因而被称为长尾分布。

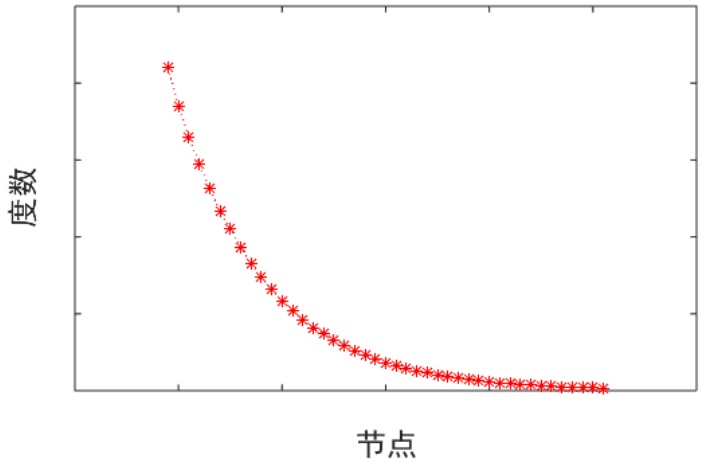


图 2.3 长尾分布（幂律分布）

2.1.3 最短路径长度和平均路径长度

节点i和节点j间的最短路径是指连接彼此的边数最少的路径，节点i和节点j间

的最短路径长度d_{ij}等于沿着最短路径经过的边的个数。在复杂网络中最短路径应用具有重要的意义，例如在互联网中，当一台计算机向另一台计算机发送数据包时采用最短路径进行传输会比其它路径更快速、更节省系统资源。

网络的平均路径长度是指其中所有节点两两之间最短路径长度的平均值，具体计算公式为：

$$L=\frac{1}{\frac{1}{2}N\left(N-1\right)}\sum_{i\geq j}d_{ij} \tag{2.4}$$

在图 2.4 所示的网络图中，平均路径长度$L=1.7$

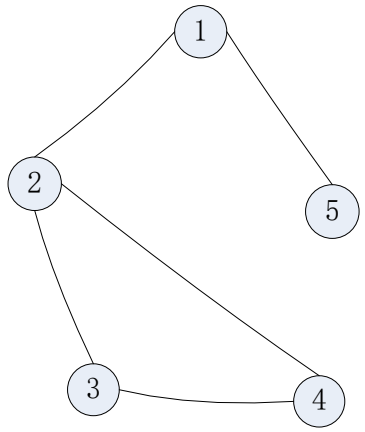


图 2.4 包含 5 个节点的网络图示例

2.1.4 聚类系数

聚类系数包括节点的聚类系数和整个网络的聚类系数。

网络中节点i的聚类系数C_i定义为：

$$C_i=\frac{E_i}{\frac{\left(k_i\left(k_i-1\right)\right)}{2}}=\frac{2E_i}{\left(k_i\left(k_i-1\right)\right)} \tag{2.5}$$

其中，E_i为节点i的邻居间实际存在的边的个数。图 2.5 中，通过公式（2.5）计算可以得到C_1=3，C_2=1，C_6=0。

网络的聚类系数描述的是网络的凝聚性，具体值是将点的聚类系数求平均数。网络的聚类系数C定义为：

$$C=\frac{1}{N}\sum_{i=1}^{N}C_i \tag{2.6}$$

图 2.5 中所示网络的聚类系数$C=\frac{1}{6}\sum_{i=1}^{6}C_i=\frac{109}{180}$。

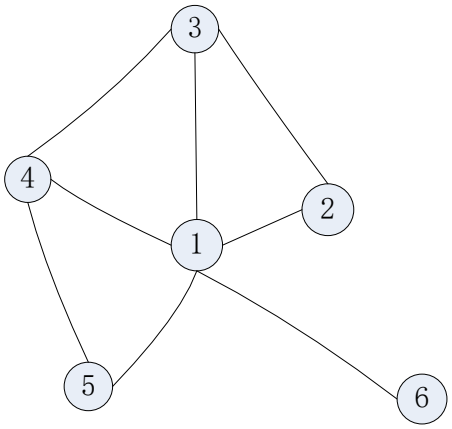


图 2.5 包含 6 个节点的网络图示例

2.1.5 社区结构和模块度

社区结构性质具体表现是，网络通常可以按照某种规律划分成很多不同的社区，各个社区的职责和拥有的成员不同，处于同一社区的成员之间通常具有相同的兴趣或是联系比较紧密，而不同社区的成员联系较为稀少。在复杂网络的产生和发展演化过程中，一些具有相同特征或爱好的主体，他们之间的联系就会较为集中，进而他们会很容易就形成一个社区。图 2.6 为 Zachary 空手道俱乐部的社区发现结果，该社区结构产生的初始原因就是节点 1 和节点 34 所代表的人员产生了利益分歧。

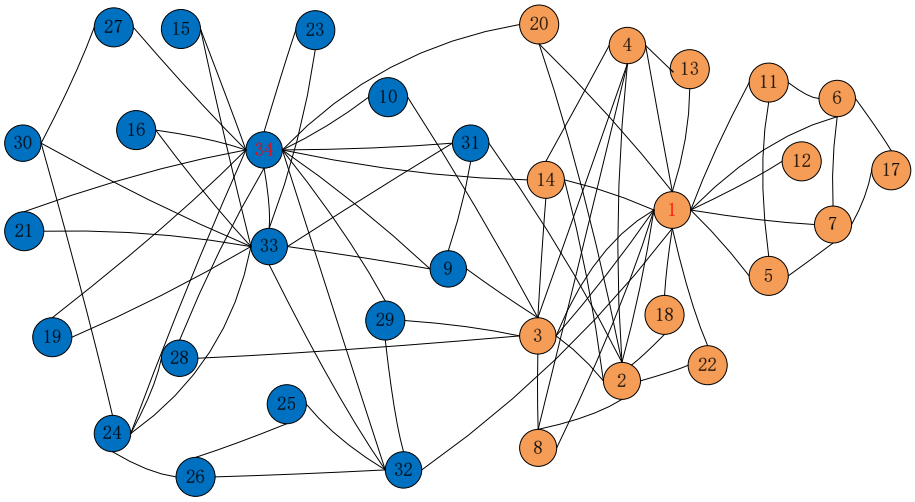


图 2.6　Zachary 社区结构示例

模块度是一种用以衡量社区划分质量的指标[14]，现已在社区发现领域得到广泛的认可和使用，定义如下：

$$Q=\sum_{i}\left(In_{ii}-Out_{i}^{2}\right) \tag{2.7}$$

其中，In_{ii} 表示在第 i 社区内部的边占网络中总边数的比例，$Out_i=\sum_j e_{ij}$ 表示一个节点在第 i 社区内部另一节点在其余社区内部的边占网络中总边数的比例。在一般

情况下，模块度Q值的大小代表着该算法的执行结果是否准确。

2.2 传统社区发现算法

现已有很多种类的社区发现算法，本节主要对下述几种类别下的经典算法进行了介绍分析。

2.2.1 基于图分割的算法

图分割是包括在并行计算、电路布局和算法设计等很多领域里的基本问题，同时也是用以解决偏微分方程和稀疏线性方程的基础技术。图分割算法的基本思想是循环迭代的将网络划分成事先设定好规模的两个社区，停止条件是这两个社区之间的边的数量达到最小，该数量即为图论中的最小割。对图 2.7 中的网络进行图分割，得到虚线左右两个社区。

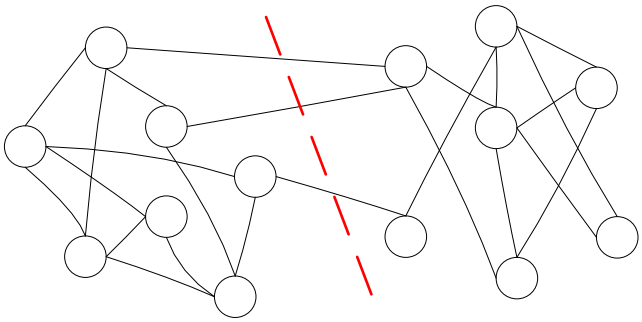


图 2.7 图分割算法示例

K-L 算法[22]和谱二分法[23]是经典的图分割算法，下面对 K-L 算法进行主要介绍。

K-L（Kernighan-Lin）算法是社区发现领域里应用较早并仍在频繁使用的算法。此算法的主要思想是不断的迭代优化一个增益函数Q，最终使Q值达到最大。其中Q表示的是社区内部边的数量与社区之间边的数量的差值。该算法的具体步骤为：

Step1 为社区指定大小；

Step2 将网络中节点随机分为两个社区C_1和C_2；

Step3 在社区C_1中随机选择一个节点i；

Step4 统计社区C_2中还未被交换过的节点，与社区C_1中的节点i进行交换，计算增益函数Q变化值ΔQ，选择其中使ΔQ最大的节点j，与节点i进行交换；

Step5 重复进行 Step4-5，直至社区C_1或C_2中所有的节点都有被换过或增益函数Q达到最大。

K-L 算法的执行速度较快，其时间复杂度为$O\left(n^2 \log n\right)$，其中n为网络中节点的

数目。但是 K-L 算法存在着一些明显的缺陷：

（1） 必须事先知晓网络中两个社区的大小。

（2）算法的执行速度与上述的 Step2 的随机划分社区结果好坏有很大关系。

（2）该算法是一种贪婪算法，迭代次数成为制约算法执行速度的重要因素。

由于上述这些缺陷，K-L 算法应用于真实的复杂网络中是比较困难的。

2.2.2 层次聚类算法

层次聚类算法是应用最广的一类方法，通过分析网络中节点间关系的强弱对网络进行划分，主要有下述两种类型：

2.2.2.1 分裂聚类算法

分裂聚类算法是首先将网络看成是一个社区，随后迭代的将相似度最低的边去掉，逐渐将这个社区割裂成若干社区，从而完成社区发现过程。整个分裂聚类算法过程可以通过树状结构图来表示，如图 2.8 所示：

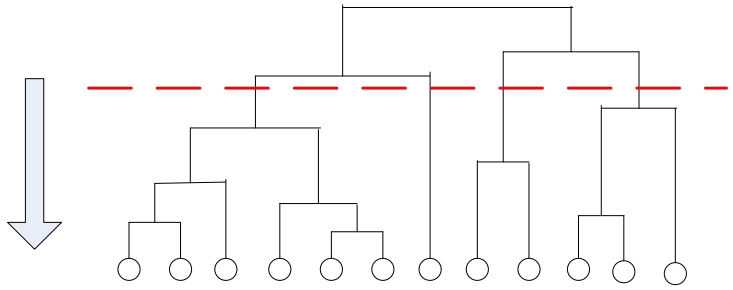
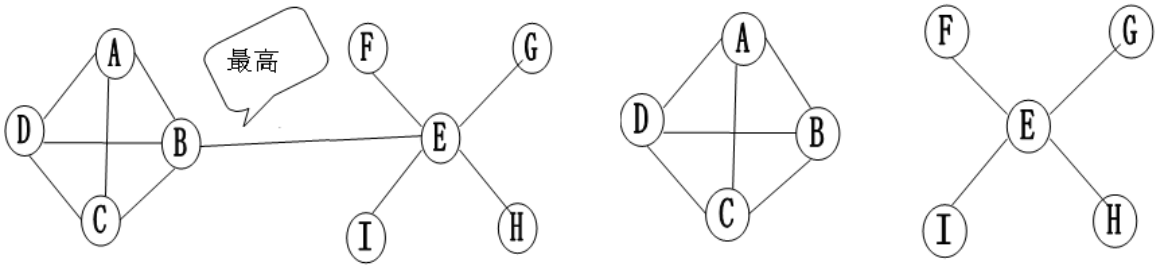


图 2.8 分裂聚类算法的树状结构图

最著名的分裂聚类算法是 GN 算法[14]。GN 算法的主要思想是：处于同一社区的节点间联系比较紧密，而不同社区间的节点联系较为稀疏，连接不同社区的边拥有着大的边介数，将这些社区间的边移除，最终完成社区发现过程。下图 2.9 为 GN 社区发现算法原理示例，移除 BE 边后，网络可以划分成两个社区。



a) 原始网络　　b) 移除介数值最高的边之后得到的社区图

图 2.9 GN 社区发现算法原理示例

GN 算法的主要步骤为：

Step1 计算并存储此时网络中所有边的边介数；

Step2 将 Step1 中计算得到的最大值对应的边移除；

Step3 计算并存储此时社区结构的模块度；

Step4 重复进行 Step1-4，直至网络中所有的边都已被去除；

Step5 输出模块度最大时的划分结果。

尽管 GN 算法的提出有着很深远的意义，但是 GN 算法的时间复杂度较高，并不适用于大规模网络。

2.2.2.2 凝聚聚类算法

与上述 2.2.2.1 描述的算法完全相反，整个凝聚聚类算法过程的树状结构图如图 2.10 所示。

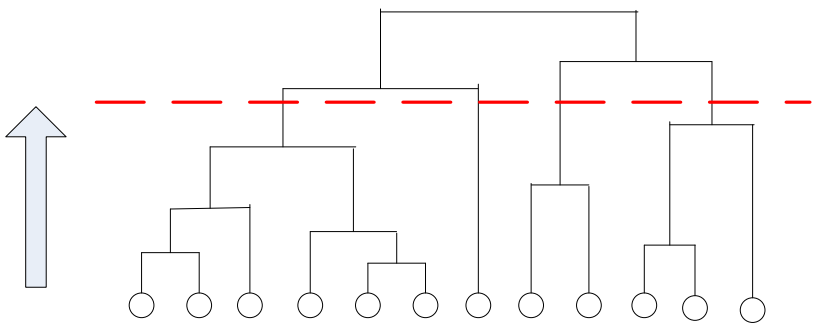


图 2.10 凝聚聚类算法的树状图

Newman 快速法是一种著名的凝聚聚类算法[24]。Newman 快速法的主要步骤为：

Step1 将网络中每个节点都作为一个独立的社区；

Step2 计算并存储此时社区结构的模块度；

Step3 将有连接的社区进行合并，若合并后模块度变化值大于 0，则保留合并；否则，去除此次合并；

Step4 重复进行 Step2-3 使得网络中只剩余一个社区。

Step5 输出模块度最大时的划分结果。

Newman 快速法的时间复杂度为$O\left(n\left(m+n\right)\right)$，效率比 GN 算法有着不少的提高，但是由于 Newman 快速法仍是基于贪婪的算法，仍会造成资源上的很大浪费。

除 Newman 快速法以外，CMN 算法[25]也是一种应用很广泛的凝聚算法，其利用数据结构中的堆结构进行计算和存储大大提高了运算效率和节省了存储资源。

2.2.3 模块度优化算法

GN 算法中提出的模块度Q自提出以来已逐渐成为许多聚类算法的停止条件，是迄今为止最知名也是最常用的社区质量衡量指标，它的大小能直接反应某一算法是否有效。模块度的这一特性，使得出现了基于此特性优化的一类算法。

2.2.2 中的 Newman 快速算法是较早提出的一种模块度优化算法，在上一小节已经介绍过了，这里不再赘述类似的在层次聚类中运用模块度优化思想的算法，如模拟退火法算法等[26]。本小节介绍一种直接寻找最优值的相关算法：EO 社区发现算法[27]。

EO 社区发现算法的主要思想是为每个节点都计算一个局部模块度q，对整体的模块度的贡献有大有小，迭代地调整每一个q最终使网络整体的模块度达到最大。对于社区i节点u的局部模块度$q_i(u)$的定义如下：

$$q_i(u) = in_i(u) - k_u out_i(u) \tag{2.8}$$

其中$in_i(u)$表示在社区i中与节点u相连的边的个数，k_u为节点u的度数，$out_i(u)$表示不在i的且与u相连的边的个数。由公式（2.7）可知，局部模块度q_i对网络整体的模块度Q的贡献可定义如下：

$$Q = \frac{1}{2M}\sum_i q_i \tag{2.9}$$

其中M为网络中边的总数。

EO 社区发现算法的主要步骤如下：

Step1 将所有节点随机分为两个部分，每一部分代表一个社区；

Step2 计算社区中各个节点对于该社区的局部模块度q（贡献度），将q最小的节点移到另一个社区中，计算模块度Q；

Step3 重复进行 Step2，使得模块度Q达到最大，并在此时删除社区之间的所有边；

Step4 对各个社区再次执行 Step1-3，直至模块度Q不再得到提高。

EO 算法比图分割算法有着比较大的改进，社区的个数可以得到自动确定。时间复杂度为$O\left(n^2 \ln n\right)$，计算效率逊于 Newman 快速法。

除了 EO 社区发现算法以外，整数规划算法[28]也是一种直接寻找模块度最优值的社区发现算法，它通过利用数学中整数规划的方法去求解网络中社区结构模块度的最大值，为社区发现的技术发展提供了新的方向。虽然基于模块度优化的算法在

社区发现领域里颇为流行，但是此类算法存在不能识别出小规模社区和很难在大规模复杂网络上得以应用的缺陷。

2.3 大规模复杂网络的社区发现算法

从前文可以看出，针对社区发现算法的研究现已取得了一些成绩，但是随着复杂网络规模的不断增大和其种类的不断增加，前面的这些方法并不能既快速又准确地对大规模网络进行社区划分。目前针对大规模网络的算法大致有三类：依靠并行平台等技术的改进算法，局部算法和减小网络规模的算法。

2.3.1 并行社区发现算法

随着对大规模复杂网络研究的需求不断增大，很多公司和高校开发出了专门处理大规模数据处理系统或编程模型，如卡内基梅隆大学提出的 GraphLab 开源图计算框架、大规模图分布式计算平台 Pregel、Spark 分布式数据分析平台和 MapReduce 编程模型等。研究人员基于这些大规模数据处理系统和编程模型，提出了一些对现有算法进行改进的并行社区发现算法，如 PD 算法[29]、模块度优化并行实现社区发现算法[30]和 DEPOLD 算法[31]等。

这些并行算法使得其计算效率与传统算法相比有较大提高，但是这类算法通常对计算机系统的硬件要求较高，再加上传统算法自身的缺陷，导致此类算法的应用性较差。

2.3.2 局部社区发现算法

由于存储和分析海量数据会花费高昂的时间和空间开销，于是出现了一些局部算法。这类算法是只针对网络中的部分节点进行社区划分。

LWP 算法[32]是一种局部的基于贪婪的社区发现算法。LWP 算法的主要思想是选取网络中的部分节点放入某个子网络中，迭代地向该子图加入或删除节点，最终使该子网络的社区结构达到最稳定。

LWP 算法定义了社区的概念：当某个子网络的 *ind* 大于 *outd* 时，该子网络就为一个社区。其中 *ind* 是指子网络的入度，*outd* 是指子网络的出度。判断某子网络 S 是否为社区可以用局部模块性 M 来定量的表示：

$$M = \frac{ind(S)}{outd(S)} \tag{2.10}$$

若 $M > 1$，则子网络 S 为一个社区。LWP 算法的主要步骤如下：

初始化：设子网络 $S = \varnothing$， $Nodes = \varnothing$

Step1 随机选取一个节点 i，并将 i 加入到 S 中，将 i 的所有邻居节点加入到 *Nodes* 中；

Step2 选择 *Nodes* 中使局部模块性 M 增加值最大的节点加入到子网络 S 中，更新 S 和 *Nodes*；

Step3 对子网络 S 中的所有节点计算去除该节点后局部模块性 M 增加值，在保证 S 连通性的基础上，选择使增加值最大的节点从 S 中删除，更新 S；

Step4 重复执行上述的 Step2-3，直至 S 中节点达到稳定。

LWP 算法解决了存储和分析大规模复杂网络会花费高昂的时间开销等问题。但是该算法存在很大的随机性，从上述步骤可以得知节点 i 的选取对算法的结果有很大影响，而且最终节点 i 不一定会出现在该社区中，进而导致该算法存在一定的盲目性。

局部社区发现算法不能完成整个网络的社区划分是这类算法的一个先天缺陷。

2.3.3 减小网络规模社区发现算法

目前，对大规模网络进行规模处理主要采取抽样和压缩两种技术，本文 3、4 章中提出的算法分别是基于抽样技术和压缩技术的大规模复杂网络社区发现算法。

2008 年，Bolnder 等针对有权网络提出了基于模块度指标的快速压缩 L-M 算法[33]。该算法通过计算模块度增益，根据模块度最大增益值将节点进行不断压缩，使模块度不断得到优化，得到原始网络的概要图，进而发现社区结构。L-M 算法由于设计简单且社区划分结果比较准确而得到了广泛的应用。

在 L-M 算法中，模块度增益 ΔQ 定义为：

$$\Delta Q=\left[\frac{\sum C+k_i^C}{2m}-\left(\frac{\sum C'+k_i}{2m}\right)^2\right]-\left[\frac{\sum C}{2m}-\left(\frac{\sum C'}{2m}\right)^2-\left(\frac{k_i}{2m}\right)\right] \tag{2.11}$$

其中，$\sum C$ 为社区 C 的内部边的权值之和；k_i 为以节点 i 所在的边的权值之和；$\sum C'$ 为只有一个节点在 C 中的边的权值之和；k_i^C 为将 i 加入到社区 C 中后，含有 i 在社区内部的边的权值之和；m 为网络中所有边权值的和。

L-M 算法主要步骤如下：

Step1 将网络中每个节点都作为一个独立的社区；

Step2 对每个节点 i，考虑 i 的邻居节点集合 N_i，将 i 加入到使该增益最大的邻居所属的社区；

Step3 将由以上步骤划分好的社区整体分别作为一个新的节点；

Step4 新形成边的权值等于相应社区间原来的所有边权值之和；

Step5 重复 Step2-4，使得增益 ΔQ 达到最大。

图 2.11 为 L-M 算法执行过程示例，其中，原始图中边的权值默认为 1，V_j^i 表示第 i 次迭代以后的第 j 个节点。通过一次迭代，V_0^0、V_1^0、V_2^0、V_4^0、V_5^0 组成形成一个新的节点 V_1^1，V_3^0、V_7^0、V_6^0 组成一个新的节点 V_3^1，V_8^0、V_9^0、V_{10}^0、V_{12}^0、V_{14}^0、V_{15}^0 组成一个新的节点 V_2^1，V_{11}^0、V_{13}^0 组成一个新的节点 V_4^1；新形成边的权值，在图 2.11 中体现为 V_1^1 与 V_2^1 间边的权值为 1，V_1^1 与 V_3^1 间边的权值为 4，V_1^1 与 V_4^1 间边的权值为 1，V_2^1 与 V_4^1 间边的权值为 3，V_3^1 与 V_4^1 间边的权值为 1。接着进行第二次迭代。最终将网络折叠为只具有两个节点 V_1^2 与 V_2^2 组成的网络。由此可知，L-M 算法能够对原始社会网络进行压缩折叠，进而达到减小网络的规模的目的。

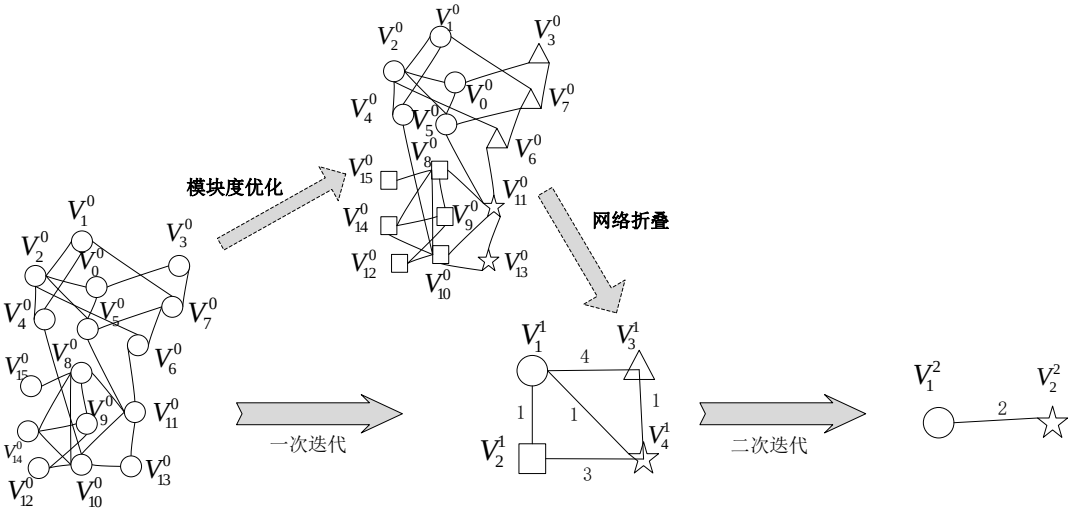


图 2.11 L-M 算法执行过程

L-M 算法使得网络规模会随着迭代次数的增加而逐渐减少，使得对于大规模网路进行社区发现成为可能。但是 L-M 算法的初始阶段仍然需要遍历存储整个网络，仍会花费高昂的时间开销和空间开销。

2.4 本章小结

针对社区发现的研究已经成为很多基础理论诸如图论、数据挖掘、社会网络分析和生物网络研究等的核心课程内容。本章首先介绍了一些与社区发现相关的基础知识，包括相关概念和数学定义公式；之后对传统的社区发现算法下的一些典型算法进行了介绍分析；最后对一些针对大规模复杂网络的社区发现算法进行了分类和介绍分析。本章内容为后续两章的提供了前期准备和理论依据。

第三章 基于抽样的大规模复杂网络社区发现算法

3.1 引言

随着近年来互联网高速发展和移动终端的普及应用，使得复杂网络的种类和规模得到了快速的发展和变化，传统的方法并不能既快速又准确地对大规模网络进行社区划分。因此，如何在保持社区发现有效性的同时，设计高效的社区发现算法显得尤为必要。

在大数据领域，可以利用抽样技术获得较少的、有代表性的样本，通过对样本进行分析，可以直接提高算法的效率。抽样过程如图 3.1 所示，通过对图 3.1 中左图进行抽样可得到右图的样本子图。

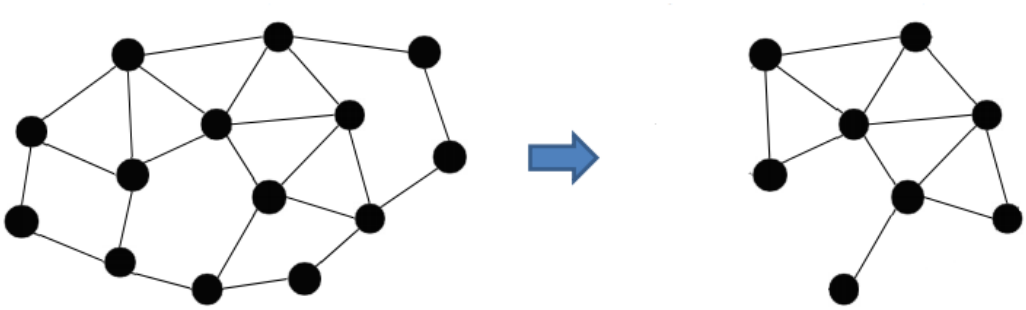


图 3.1 抽样过程

本章提出一种基于抽样技术的算法。该算法首先利用基于度的随机游走技术对整体网络进行抽样得到子图，然后将基于概要的发现算法应用到此子图上进行社区发现，之后获得初始社区，最后依据已有社区结构与未抽样的节点的相似度迭代式地将剩余节点进行划分。

通过与另外的算法进行比较分析，表明了该算法能够在保证准确性的同时提高计算效率。

3.2 基于抽样的大规模复杂网络社区发现算法

研究的对象为无向无权网络 $G=(V,E)$，其中，$V=\{V_i|i=1,...N\}$ 为节点的集合，$E=\{e_{ij}|V_i,V_j\in V\}$ 为边的集合，$N=|V|$，$M=|E|$，即 N 为节点的总数量，M 为边的总数量。

3.2.1 基于随机游走的偏采样抽样子算法

传统的随机游走抽样（Random Walk Sampling）就是将每次随机游走所经过的不

重复节点作为样本节点，一次随机游走会经过一系列的迭代过程，每次的迭代中，当前访问节点的所有未被访问过的邻居节点有相同的概率被选择为下一次迭代中的访问节点。

因此，由于传统的随机游走抽样每次在选择邻居节点时，每个邻居节点都有相同被抽中的概率，并不能够获得真正有代表性的节点，甚至会获得一些邻居节点很少的稀疏节点。所以本文采取以下模式进行随机游走抽样，当前节点$\boldsymbol{V_u}$与其任一邻居$\boldsymbol{V_v}$相连的边$\boldsymbol{e_{uv}}$被访问的概率计算如下：

$$P(e_{uv})=\frac{k_v}{\sum_{V_i\in N(V_u)}k_i} \tag{3.1}$$

其中k_v为节点V_v的度，$N(V_u)$为节点V_u的邻居节点集合。

研究表明，复杂网络中节点的度数越大，该节点成为社区代表的可能性越大[34]。因此，相比于传统的随机游走抽样技术，基于以上的抽样策略，更能获得有代表性的子图。

本章采取 p 次基于随机游走的偏采样子算法，得到子图G'。一次偏采样子算法描述如下：

算法 3.1 一次基于随机游走的偏采样子算法

输入：原始网络图$G=(V,E)$，游走最大步数k

输出：子图$G'=(V',E')$，其中$V'\subset V$，$E'\subset E$

Step1 初始化$G'=(V',E')$，$V'=\varnothing$，$E'=\varnothing$；

Step2 计算各个节点的度数占网络中总度数的概率，根据此概率进行偏采样，选取一个初始节点V_1，更新$V'=V'\cup\{V_1\}$，将当前访问节点记为$V_{\text{current}}=V_1$；

Step3 计算节点V_{current}的邻居集合，判断此集合中节点是否已经访问过，将未被访问过的节点集合记为$N'(V_{\text{current}})$；

Step4 判断$N'(V_{\text{current}})$是否为空，若为空集，则停止此次游走；若不为空，则按照公式（3.1）计算节点V_{current}与其未被访问过邻居的连边被访问的概率，根据概率进行偏抽样选取V_{current}的一个邻居节点V_{x}，并更新$V'=V'\cup\{V_{\text{x}}\}$，$E'=E'\cup\{e_{\text{xcurrent}}\}$，当前访问节点$V_{\text{current}}=V_{\text{x}}$；

Step5 重复进行 Step3-4，直至随机游走的步数达到k或无法获得新的节点。

例 3.1 如图 3.2 所示，假设原始网络如左图所示，共包括 12 个节点。经过 5 次随机游走抽样，每次游走最大步数为 3（第 1 次，W→R→P；第 2 次，M→N→P；

第 3 次，N→Q→M；第 4 次，W→P→N；第 5 次，Y→W），得到包括 7 个节点的子网络，如右图所示。

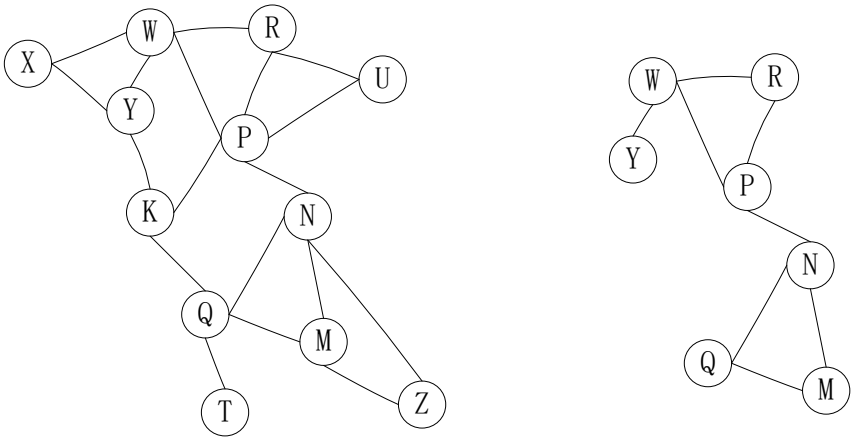


图 3.2 随机游走抽样示例

在得到子图$G'=(V',E')$后，统计G'中所有节点和边被访问的次数。利用互信息计算 p 次抽样后的子图中每条边的权值，定义如下：

$$W(e_{ij})=\log\frac{P(e_{ij})}{P(V_i)P(V_j)} \tag{3.2}$$

其中：

$$P(e_{ij})=\frac{CountEdge(e_{ij})}{\sum CountEdge(e_{ij})} \tag{3.3}$$

$CountEdge(e_{ij})$为子图中的边e_{ij}在 p 次抽样中被经过的次数。

$$P(V_i)=\frac{CountNode(V_i)}{\sum CountNode(V_i)} \tag{3.4}$$

$CountNode(V_i)$为子图中点V_i在 p 次抽样中被经过的次数。

3.2.2 基于标签传播的扩充子算法

在对子图G'采用 2.3.3 中描述的 L-M 算法后，得到初始划分结果。然而此结果没有并没有包含G 中的所有节点，因此须将G 中剩余的节点进行划分。

基于标签传播的扩充子算法的根据可以描述为：在对G'进行社区发现后，G'中的每个节点都对应有一个社区标签，因为G'中都是有代表性的节点，根据子图G'中节点的标签，为G 中未被抽样的节点进行社区划分，进而完成所有节点的社区划分。

算法 3.2. 基于标签传播的扩充子算法

输入：原始网络图$G=(V,E)$，子图$G'=(V',E')$，初始社区发现结果

$C=\{C_1,C_2,...,C_t\}$

输出：整体网络的社区发现结果

Step1 根据初始社区发现结果，为G'中的所有节点赋予标签，记为$Label(V_x)$，例如节点V_i在初始社区发现结果C_t中，则$Label(V_i)=t$；

Step2 计算子图中各个节点的度数占子图中节点总度数的概率，根据此概率对子图中节点进行偏采样，选取一个节点$V_i \in V'$；

Step3 将V_i的邻居节点集合$N(V_i)$中没有标签的节点记为$N^{un}(V_i)$，将$N^{un}(V_i)$中节点赋予与节点V_i相同的标签，即$Label(V_i)$；

Step4 更新V'，其中$V'=(V'-\{V_i\})\cup N^{un}(V_i)$；

Step5 重复进行 Step2-4，直至网络G中所有的节点都被赋予了标签。

例 3.2 图 3.3 左图中阴影区域表示抽样得到的子网络，假设对子网络采用 L-M 算法之后得到C_1和C_2两个社区，其中$C_1=\{W,R,Y,P\}$，$C_2=\{Q,N,M\}$。根据算法 3-2 首先将图 3.3 左中所有节点赋予标签值，如$Label(P)=1$，$Label(M)=2$；在 Step2 步骤选中V'中节点 P，而$N^{un}(P)=\{U,K\}$，所以为节点 U 和 K 赋予标签节点 P 的标签，即$Label(U)=1$和$Label(K)=1$，同时更新子图节点集合为$V'=\{W,R,Y,Q,N,M,U,K\}$；接着选中V'中节点 M，$N^{un}(M)=\{Z\}$，所以$Label(Z)=2$，同时更新子图节点集合为$V'=\{W,R,Y,Q,N,U,K,Z\}$；同理根据以上步骤得到$Label(X)=1$，$Label(T)=2$。最终整个网络的社区发现如图 3.3 中右图所示。

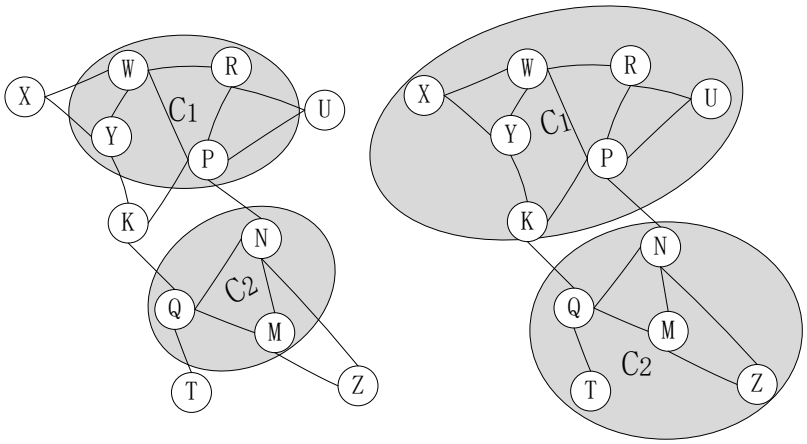


图 3.3 剩余节点扩充

3.2.3 基于抽样的社区发现算法

基于抽样的社区发现算法主要分为三个阶段：第一阶段是对原始网络多次采用基于随机游走的偏采样子算法；第二阶段是对子图采用 L-M 算法，得到抽样网络的社区发现结果；第三阶段是将已有社区结果传播到剩余节点，完成社区发现结构的

扩充。

输入：原始网络图$G=(V,E)$，偏采样子算法执行次数p，每次偏采样子算法中最大步数k

输出：社区发现结果

Step1 执行p次最大游走步数为k的算法 3.1 过程，得到子图$G'=(V',E')$；

Step2 根据子图G'中各节点和各条边被访问过的概率，根据公式（3.2）计算子图中边的权值，对子图G'执行 L-M 算法，得到子图的划分结果；

Step3 执行算法 3.2 过程，为网络G中剩余节点赋予标签，即将G中剩余节点扩充到初始社区发现结果中，得到整体网络的社区发现结果。

3.3 实验分析

实验采用真实网络数据对本章提出的算法进行检验。通过该数据就 L-M 算法[33]、CBCD 算法[35]和本章提出的算法进行了比较分析。

本章所比较的 CBCD 算法[35]中核心图是根据原文的策略进行选取，为了比较的公平性，CBCD 算法在进行核心图社区发现时采取 L-M 算法而不是 GN 算法；由于本文研究对象为无向网络，因此 L-M 算法中的模块度使用公式（2.7）计算；在本实验中，本章提出算法中随机游走抽样次数$p=2\sqrt{n}$，每次游走最大步数$k=20$。实验环境为：32 GB 内存， Intel (R) Xeon E5-2665 处理器，2.4 GHz，Windows Server 2008 操作系统。

3.3.1 数据集

本章采用 5 个较大的真实数据集进行测试，具体参数如表 3.1 所示。

表 3.1 实验数据集

数据集	节点数	边数	来源
Ca-HepPh	12008	237010	高能物理论文作者网络[36]
Ca-AstroPh	18772	396160	天文物理学类论文作者网络[36]
Ca-CondMat	23133	186936	凝聚态类论文作者网络[36]
Cit-HepTh	27770	352807	高能量物理论文引用网络[37]
Cit-HepPh	34546	421578	高能量物理现象引用网络[37]

3.3.2 度量指标

采用模块度[14]作为有效性评价指标来对实验结果进行评估，模块度定义如公式（2.7）所示。模块度Q越高，实验结果越能代表真实的社区结构。采用运行时间（秒）

作为高效性评价指标对实验结果进行评估。

3.3.3 实验结果与分析

在对以上 5 个数据集采用 L-M 算法、CBCD 算法和本文提出算法分别运行进行 50 次，平均指标值分别如表 3.2 和表 3.3 所示。其中“—”表示算法内存溢出或 48 h 未运行出结果。

表 3.2 模块度比较

数据集	L-M	CBCD 算法	本章提出算法
Ca-HepPh	0.34	0.23	0.30
Ca-AstroPh	0.31	0.21	0.23
Ca-HepPh	0.32	0.23	0.27
Cit-HepTh	0.59	0.02	0.37
Cit-HepPh	0.68	—	0.49

表 3.3 运行时间（秒）比较

数据集	L-M	CBCD 算法	本章提出算法
Ca-HepPh	2257.3	1333.3	110.9
Ca-AstroPh	15900.5	751.0	275.8
Ca-HepPh	4676.1	358.7	169.1
Cit-HepTh	110717.5	1219.2	827.9
Cit-HepPh	63648.8	—	284.48

由表 3.2 可知，通过模块度指标进行评价，在有效性方面本文提出的算法与 L-M 算法相差不多，但稍优于基于抽样策略的 CBCD 算法；由表 3.3 可知，通过算法运行时间进行比较分析，在高效性方面本文提出的算法明显优于其他两种社区发现算法。

3.4 本章小结

本章提出了一种基于抽样的大规模网络社区发现算法，该算法首先对原始网络进行多次的随机游走抽样，根据得到的子图及子图中节点和边出现的概率计算各条边的权值，对子图执行 L-M 算法，然后将网络中为被抽样的节点扩充到初始结果中。通过与另外的算法进行比较分析，表明了该算法能够在保证准确性的同时提高计算效率。

第四章 基于压缩的大规模复杂网络社区发现算法

4.1 引言

由于传统的社区发现算法大多都需要得知整个网络的拓扑结构才能进行社区划分，但是这会耗费很多的时间和空间开销，因此研究人员在对复杂网络进行处理使规模变小这个角度进行了一些方法研究，抽样技术就是其中的一种方法。从上章中我们可以看出利用抽样技术可以大大减小复杂网络的规模，但是抽样技术只是单纯从复杂网络总体中选取一部分样本来代替整个网络，而不是包含整个网络的所有信息，导致算法本身会给划分结果带来很大的误差影响。从 3.3.3 小节的实验对比中可以看出利用抽样技术的算法虽然在高效性方面较传统算法有很大提高，但是在有效性方面与其他算法对比仍有差距。

与抽样技术类似，压缩技术也是一种使大规模网络减小的处理技术。压缩技术主要通过对大规模网络进行压缩得到一个能够反应总体结构的简明概要图，如 MCDTM 模型算法[38]。对大规模网络进行概括之后，基于得到的概要图能够更高效地进行存储，分析和可视化。压缩过程如图 4.1 所示，通过对左图采用概要技术可以得到右图所示的概要图。因此，此项技术能够大大提高算法的效率。与此同时，压缩技术并不如抽样技术般只是对网络中的部分节点进行处理，好的概要图包含了原始网络中所有节点信息，很好的改善了抽样技术信息缺失的缺陷。

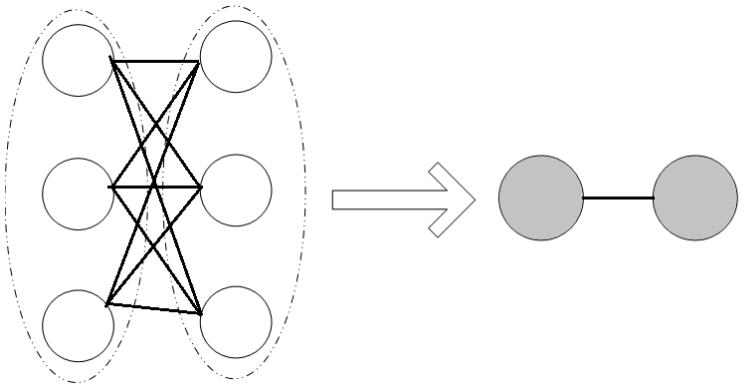


图 4.1 压缩过程

为了保证高效发现社区结构的同时提高准确性，本章提出了基于压缩技术的算法。该算法首先根据复杂网络中的普遍特性——长尾特性将网络中节点不断进行压缩折叠，然后对压缩后的概要图进行核心节点选取并将核心节点组成初始社区，最后完成剩余节点和已被压缩节点的社区划分。

4.2 基于压缩的大规模复杂网络社区发现算法

研究的对象为无向无权网络$G=(V,E)$，其中，$V=\{V_i|i=1,...N\}$为节点的集合，$E=\{e_{ij}|V_i,V_j\in V\}$为边的集合，$N=|V|$，$M=|E|$，即$N$为节点的总数量，$M$为边的总数量。

4.2.1 压缩子算法

近年来的深入使得人们发现在许多复杂网络中绝大多数节点的度是比较小的，只存在极少数度值较大，以 Facebook 朋友关系为例：只有少数的用户有着超过 2000 人的朋友数量，而大部分用户的朋友数量都不足 200 人。从图 2.3 中可以看出，这些网络的度分布图有一个长长的尾巴。图 4.2 是一个具有长尾特性的网络图示例，度数很小的边缘节点占有很大比例。

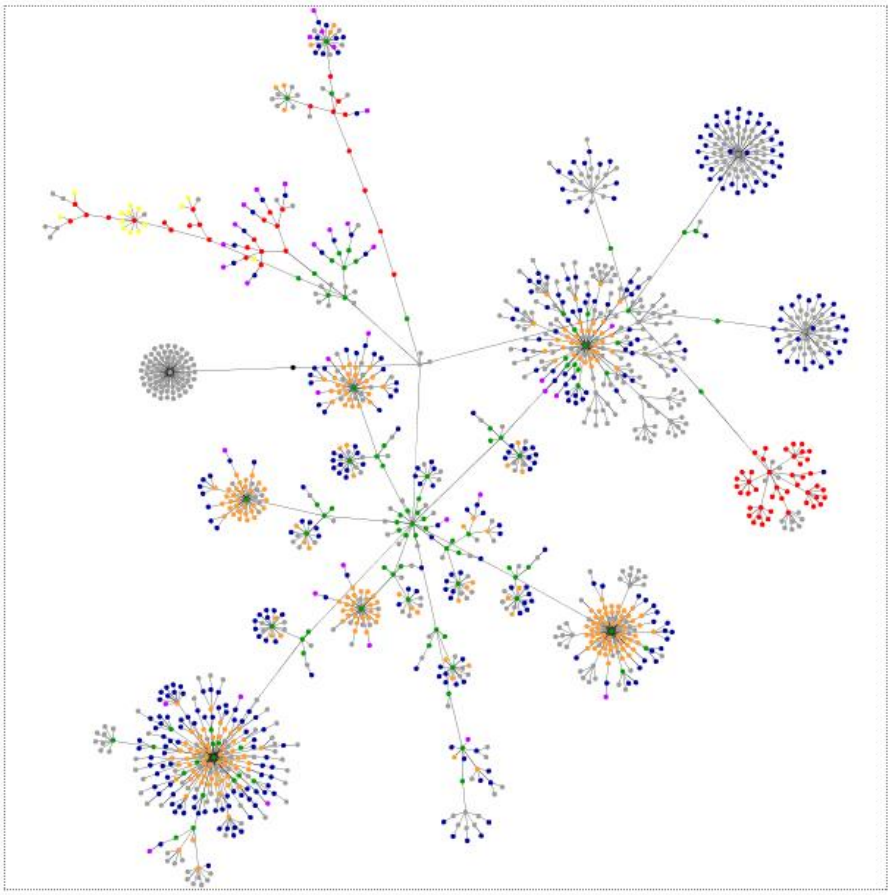


图 4.2 长尾特性网络图示例

复杂网络中度数为 1 的边缘节点只有一个邻居节点，根据社区结构的定义，此边缘节点会和它的唯一邻居节点处于同一社区结构中。由于复杂网络的长尾特性，网络中存在着大量的度数很小的边缘节点，因此可以将这些边缘节点压缩到其邻居节点中，在不损失网络的信息基础上达到简化社区发现对象规模的目的。本章提出

的压缩子算法的主要思想如下：对网络中度数为 1 和 2 的节点进行压缩折叠，将其压缩到度数最大的邻居节点中，并统计压缩后网络中每个剩余节点中包含的被压缩节点个数。

定义：节点u的质量$Quality_u$，为经过压缩步骤之后u中所包含的被压缩过的节点个数。

节点u的压缩列表$List_u$，为经过压缩步骤之后中u所包含的被压缩过的节点的详细信息。

算法 4.1 压缩子算法

输入：原始网络图$G=(V,E)$

输出：压缩后网络图$G'=(V',E')$，其中$V'\subset V$， $E'\subset E$

Step1 初始化$G'=(V',E')$，$V'=\varnothing$， $E'=\varnothing$， $Quality=1$， $List=\varnothing$；

Step2 计算统计网络中节点的度数k，将k为 0、1、2 的节点分别放到S_0、S_1、S_2中；

Step3 在集合S_1中随机选取一个节点u，将节点u压缩到其邻居节点v中，并置$Quality_u=0$，$Quality_v=Quality_v+1$，$k_v=k_v-1$，$List_v=List_u\cup List_v$，$List_u=\varnothing$， 根据k_v的大小，更新S_0、S_1、S_2；

Step4 重复执行 Step 3，直至$S_1=\varnothing$；

Step5 在集合S_2中随机选取一个节点 p，将节点 p 压缩到其度数最大的邻居节点 q 中，并置$Quality_p=0$， $Quality_q=Quality_q+1$， $k_q=k_q-1$， $List_q=List_p\cup List_q$，$List_q=\varnothing$，根据k_q的大小，更新S_0、S_1、S_2；

Step6 重复执行 Step 3-5，直至$S_1=\varnothing$且$S_2=\varnothing$；

Step7 输出压缩后网络图$G'=(V',E')$。

例 4.1 基于长尾特性的压缩子算法过程示例如图 4.3-4.4 所示。在含有 130 个节点的图 4.3 中的网络上执行算法 4.1 后，得到图 4.4 中所示的剩余 69 个节点的压缩网络图，压缩比率达到 47%。其中，节点 114 的$Quality_{114}=23$，节点 56 的$Quality_{56}=7$。

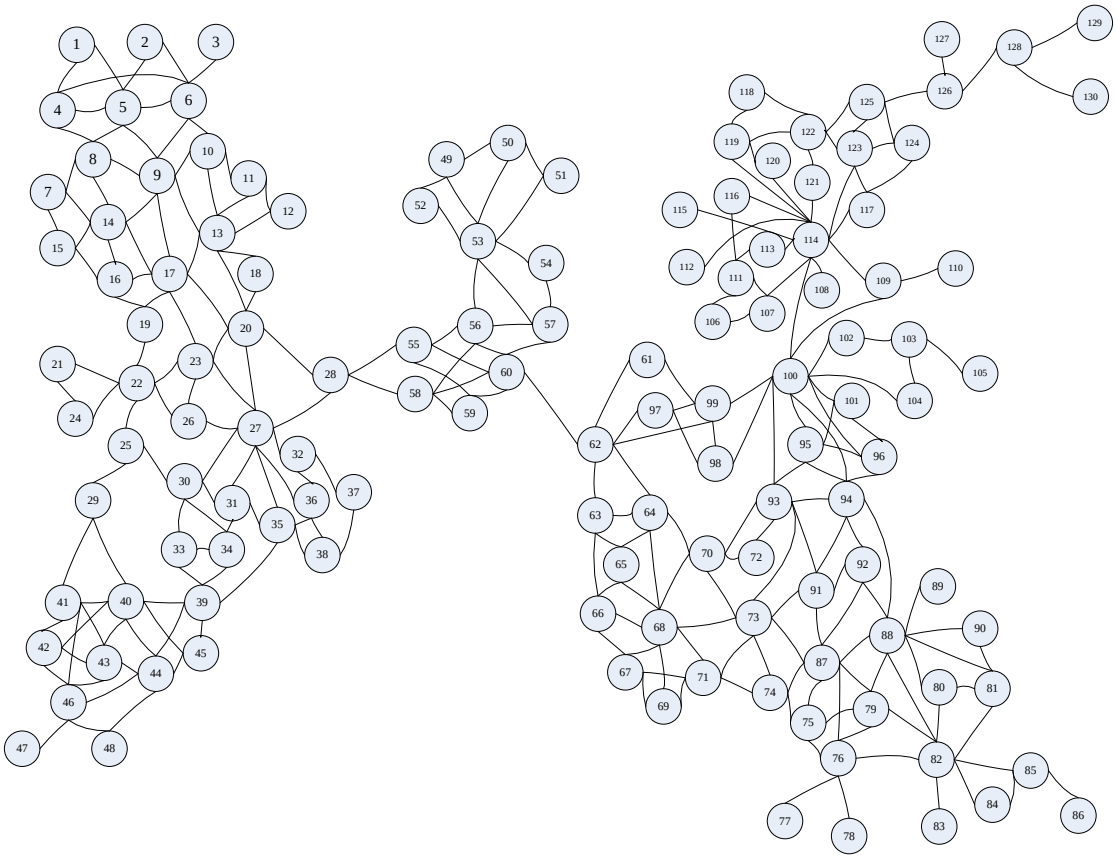


图 4.3 压缩前网络图

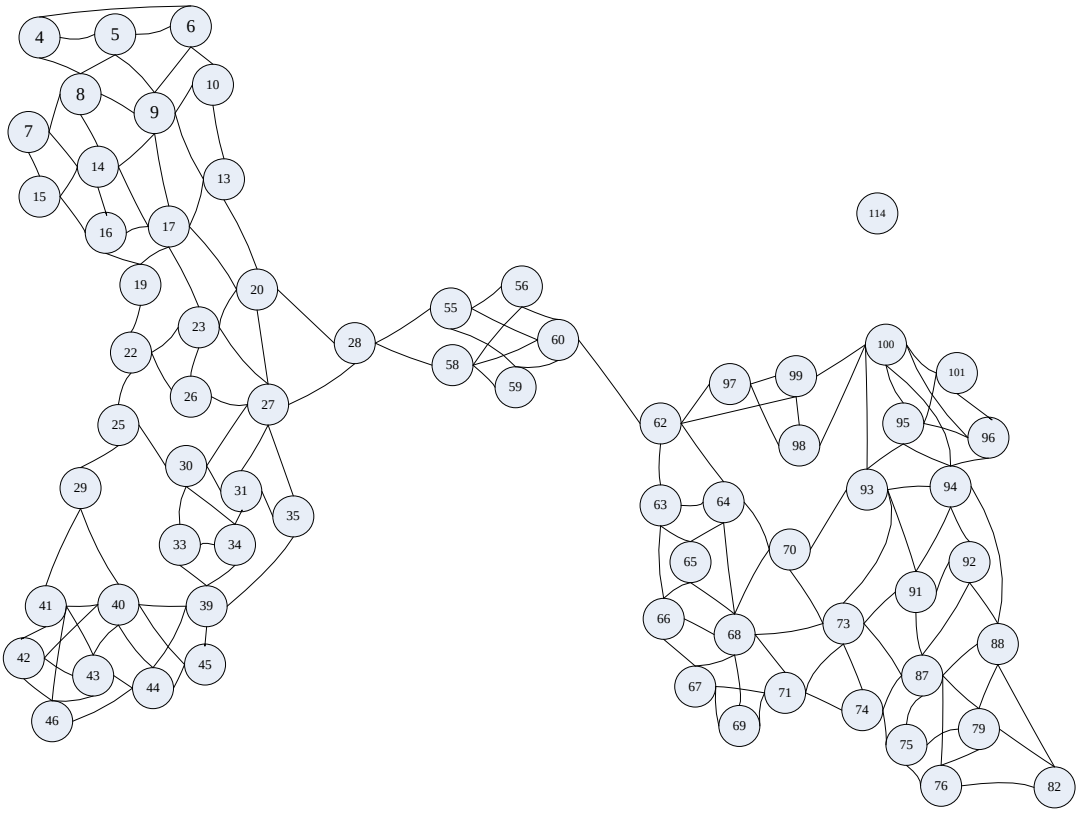


图 4.4 压缩后网络图

4.2.2 基于质量与度的社区发现算法

聚类分析与社区发现定义类似，都是将相似的或者是联系紧密的实体分成多个类，处于同一个类中的实体相似度较高，处于不同类中的实体差异性较大。因此可以利用聚类的方法进行社区研究，使划分聚类的方法思想应用于社区发现领域中。

划分聚类的主要思想是首先利用一些具体方法来选取某些实体作为类中心，之后迭代地优化变更这些类中心，使其它实体到周围最近的类中心的距离平均值达到最小。因此选取合适的初始类中心对整体算法的时效和最终的算法结果有着至关重要的影响。

2014 年 Alex 和 Alessandro 在《科学》杂志上发表的论文中，通过定义实体的密度这一概念，提出了一种简洁且有效的选取初始类中心的方法[39]。他们认为类中心拥有着较大的密度值，因此可以通过比较不同节点的密度值大小来选取初始类中心，并对密度值做了以下描述说明。

实体 i 的密度 ρ_i 定义如下：

$$\rho_i = \sum_j X\left(d_{ij} - d_c\right) \tag{4.1}$$

其中 $X\left(x\right) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases}$，$d_{ij}$ 为实体 i 与 j 间的距离，d_c 为一个超参数。该密度的具体含义表示数据中与 i 的距离小于 d_c 的实体数目。

实体 i 到高密度实体的距离 σ_i 定义如下：

$$\sigma_i = \min_{j:\rho_j > \rho_i}\left(d_{ij}\right) \tag{4.2}$$

高密度实体 i 到其它高密度实体的距离 σ_i 定义如下：

$$\sigma_i = \max_j\left(d_{ij}\right) \tag{4.3}$$

基于上述定义，该聚类算法的主要过程如下：

Step1 计算数据中所有实体的密度 ρ；

Step2 对各实体的密度进行降序排序；

Step3 计算排序后各实体到比其密度大的实体的距离 σ；

Step4 选择 ρ 和 σ 都较大的实体作为初始类中心；

Step5 将数据中其它实体根据与各类中心的距离进行划分。

该初始类中心的选择策略避免了将孤立点选为初始类中心等其余划分聚类算法

经常面临的问题，且可以识别出各种形状的聚类结果，有着很好的应用前景。复杂网络的社区发现就可以利用该算法的思想首先选取好的有代表性的节点作为初始社区中心，再进行剩余节点划分，进而达到社区划分的目的。

随着大数据时代的到来，海量的数据使得求解节点间距离的问题变得愈加困难，计算效率和所需的存储空间都受到了很大影响。所以此距离并不能直接适用上述[39]中的算法，需要根据复杂网络的特性进行改进。所以本章提出了更适合于复杂网络特性的基于质量与度的社区发现算法。

与传统聚类类似，选取合适的社区（类）中心对算法有着很大的影响。从上一小节可以得出，在大规模网络进行压缩折叠之后，每个节点都包含着一个属性值——质量*Quality*，可以认为，这一属性值表示的是该节点在原始网络中的代表性。同时在网络中节点的度数能够代表该节点的局部影响力，度数越大越有可能是关键节点。因此，基于质量与度的社区发现算法通过综合局部特征和代表性这两个判断指标来确定初始类中心。

算法 4.2 基于质量与度的社区发现算法

输入：压缩网络图$G'=(V',E')$，每个节点的质量*Quality*与压缩列表*List*

输出：整体网络的社区发现结果

Step1 计算压缩网络中各个节点的度k'；

Step2 绘制度k'和质量*Quality*的对应决策图，在此图中选取初始社区中心；

Step3 为选取的社区中心分配社区标签，并将其的一级邻居节点分配相同的社区标签；

Step4 对网络中剩余的无标签的节点按照度k'进行降序排序，查找每个节点的邻居节点，将该节点标记为与其邻居节点中出现最多的标签；

Step5 对已有标签节点包含的压缩列表*List*中的节点分配一致的标签。

例 4.2 在对图 4.4 中剩余的 69 个节点进行基于质量与度的社区发现过程如图 4.5-4.8 所示。在对网络中的质量*Quality*和度k'计算之后，图 4.5 为对应绘制的决策图，选择节点 100、27 和 114 作为初始社区中心。初始社区中心在网络中的位置如图 4.6 所示，对压缩后网络进行标签传播得到图 4.7 所示的社区发现结果。对压缩列表中的节点进行标记之后得到结果如图 4.8 所示。

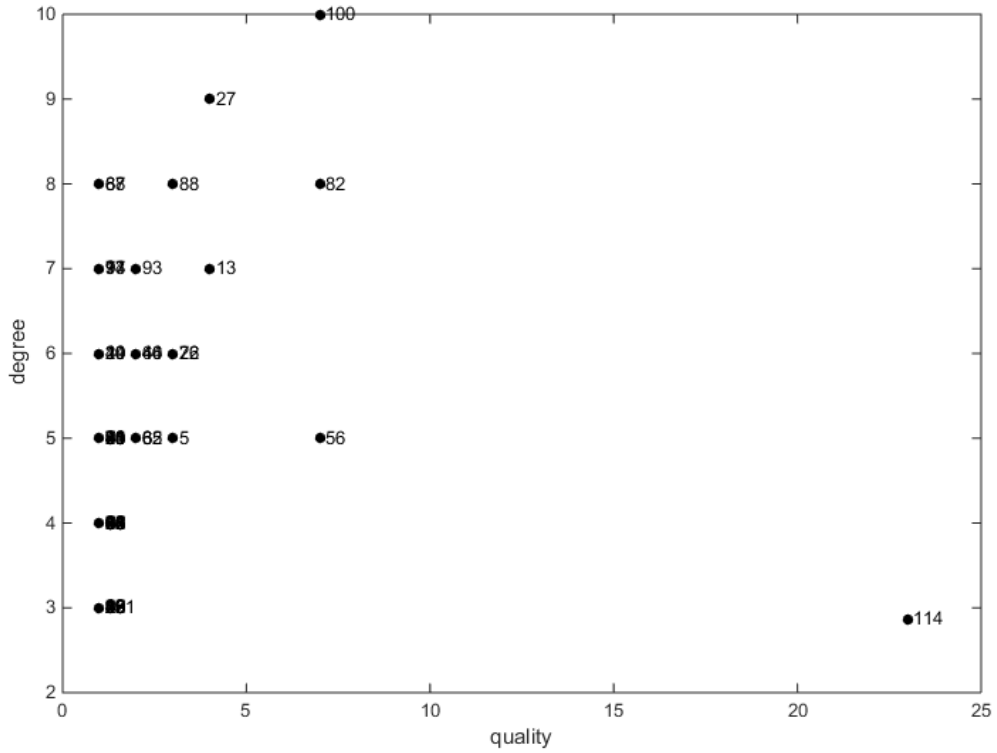


图 4.5 度数与质量的决策图

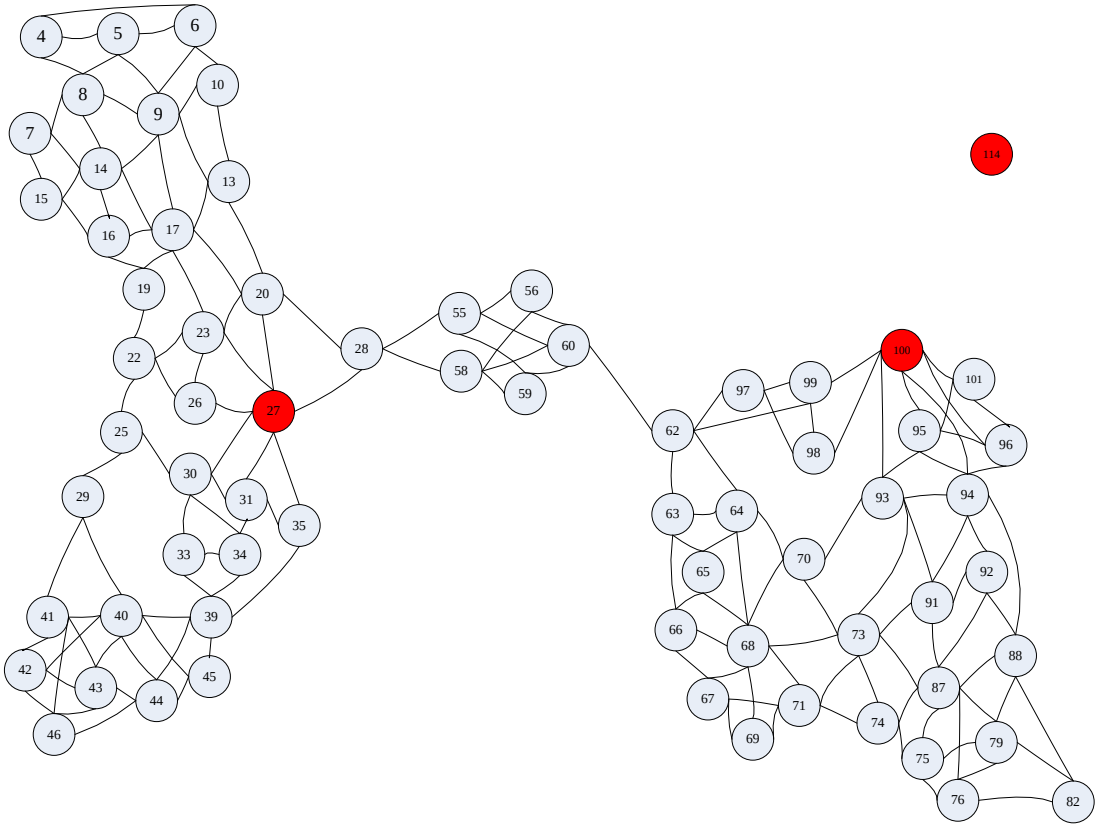


图 4.6 初始社区中心在网络中的位置

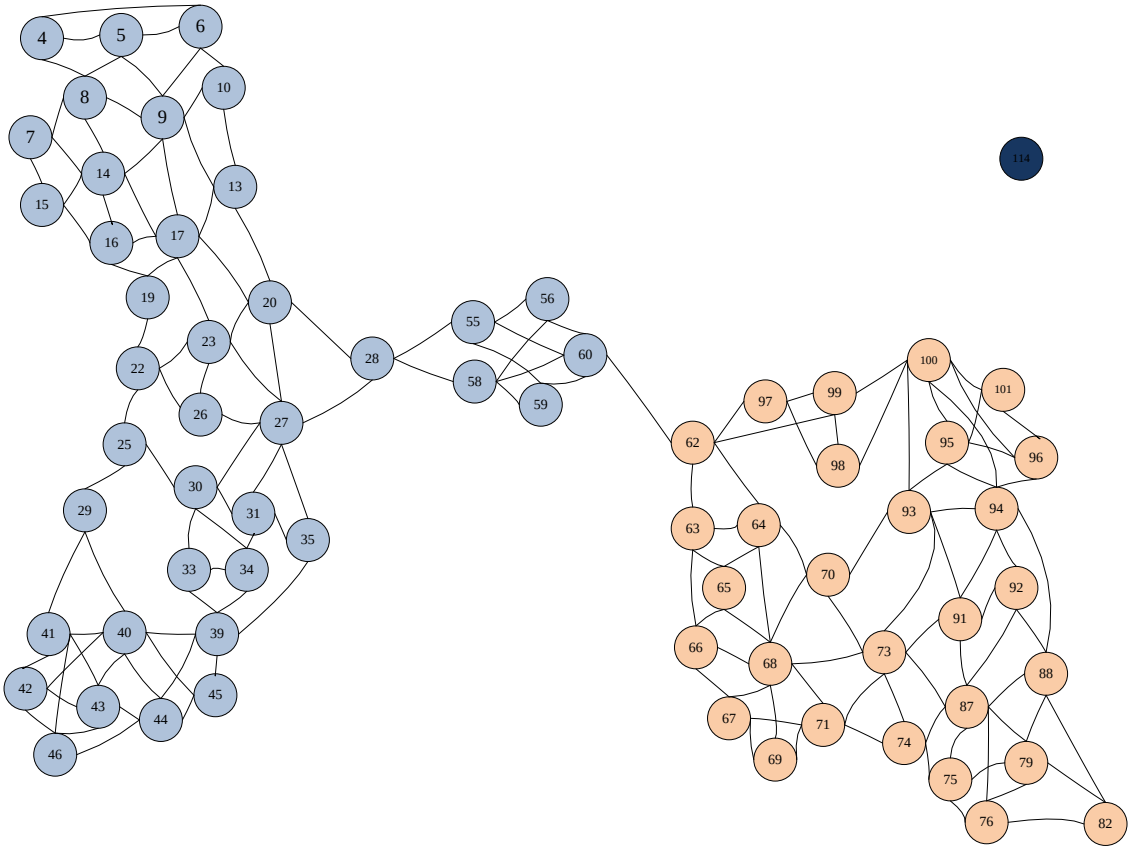


图 4.7 初始社区发现结果

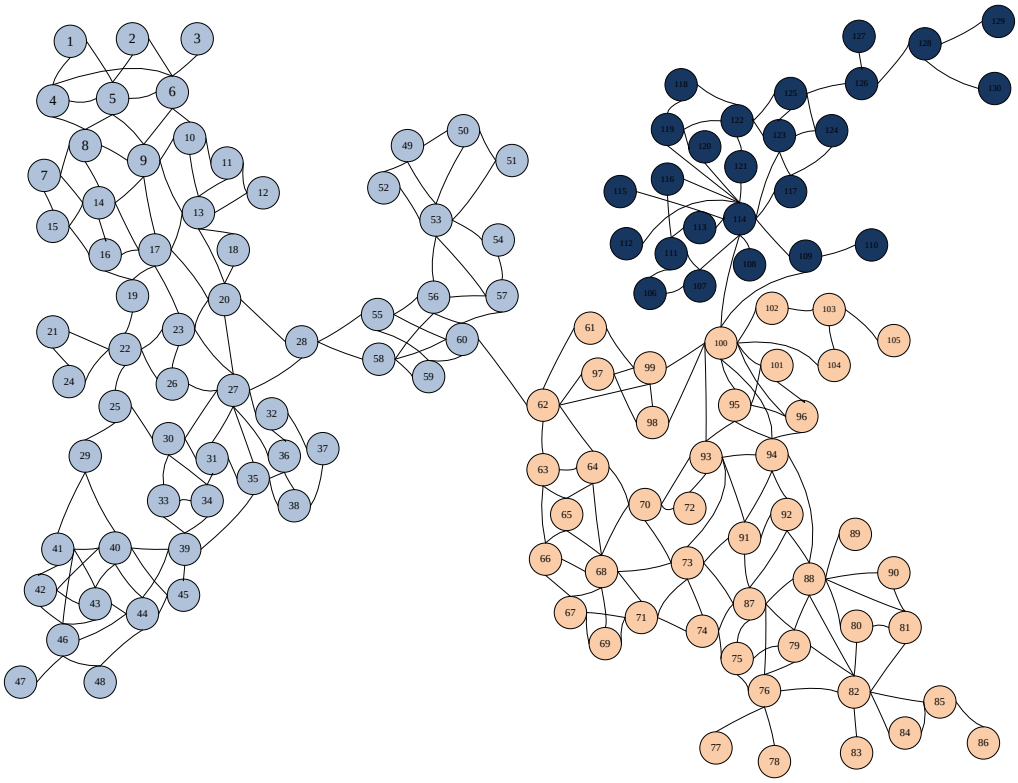


图 4.8 整体网络发现结果

4.2.3 基于压缩的社区发现算法

本文提出的算法可以分为三个阶段：第一阶段是大规模复杂网络的压缩折叠过程；第二阶段是根据压缩网络中节点的度数和质量来获得类中心并完成压缩网络的社区划分；第三阶段是将压缩网络中节点中所包含的被压缩节点进行赋予标签。其中二三阶段在算法 4.2 中同时得以实现。

输入：原始网络图$G=(V,E)$

输出：社区发现结果

Step1 对原始网络图$G=(V,E)$执行算法 4.1，得到压缩网络图$G'=(V',E')$；

Step2 对压缩网络图$G'=(V',E')$执行算法 4.2，得到原始网络的社区发现结果。

4.3 实验分析

实验采用真实网络数据对本章提出的算法进行检验。通过该数据就 L-M 算法[33]、mscd_afg 算法[40]、上章提出的基于抽样的算法和本章提出的算法进行了比较分析。

本章的实验环境为：128 GB 内存， Intel (R) Xeon E5-2650 处理器，2.6 GHz，Windows Server 2008 操作系统。

4.3.1 数据集

本章采用 6 个较大的真实网络数据集来对本章所提算法的有效性和高效性进行测试，数据集的具体参数如表 4.1 所示。

表 4.1 实验数据集

数据集	节点数	边数	来源
Ca-CondMat	23133	186936	凝聚态类论文作者网络[36]
Email_Enron	36692	183831	Enron 电子邮件网络[41]
Soc-Epinions1	75879	508837	消费者在线评论网络[42]
Email-EuAll	265214	420045	EuAll 电子邮件网络[43]
com-DBLP	317080	1049866	DBLP 作者合作网络[44]
WikiTalk	2394385	5021410	Wikipedia 讨论网络[45]

4.3.2 度量指标

采用压缩比率，即被压缩的节点占网络中总节点的比重，来对本章中所提出的压缩子算法进行评估。与第三章相同，本章采用模块度[14]和运行时间分别作为有效性评价指标和高效性评价指标来对整体算法的实验结果进行评估。

4.3.2 实验结果与分析

对以上 6 个数据集运行进行 50 次压缩子算法 4.1，得到的压缩比率如表 4.2 所示。

表 4.2 压缩比率结果

数据集	节点数	压缩比率
Ca-CondMat	23133	24.61%
Email_Enron	36692	39.49%
Soc-Epinions1	75879	66.62%
Email-EuAll	265214	88.08%
com-DBLP	317080	35.64%
WikiTalk	2394385	88.51%

由表 4.2 可知，大规模数据集经本章算法压缩折叠之后规模得以很大减少。在对各数据集采用 L-M 算法、mscd_afg 算法、上章提出基于抽样的算法和本章提出算法分别运行进行 50 次，模块度和运行时间的平均指标值如表 4.3、表 4.4 所示。其中“—”表示算法内存溢出或 200h 未运行出结果。

表 4.3 模块度比较

数据集	L-M	mscd_afg	基于抽样的算法	本章提出算法
Ca-CondMat	0.32	0.37	0.27	0.33
Email_Enron	0.30	0.33	0.20	0.31
Soc-Epinions1	0.29	0.38	0.17	0.35
Email-EuAll	—	0.69	0.46	0.66
com-DBLP	—	—	0.66	0.66
WikiTalk	—	—	—	0.36

表 4.4 运行时间（秒）比较

数据集	L-M	mscd_afg	基于抽样的算法	本章提出算法
Ca-CondMat	3478.2	250.6	143.2	44.3
Email_Enron	11229.2	1034.3	431.6	93.6
Soc-Epinions1	153380.3	7351.5	1991.4	384.1
Email-EuAll	—	43335.3	9886.8	6859.7
com-DBLP	—	—	14324.1	10735.2
WikiTalk	—	—	—	560997.5

通过表 4.3 模块度指标进行评价，在有效性方面本文提出的算法与 L-M 算法、mscd_afg 算法这两种大规模社区发现算法相差不多，且比上章所提出的基于抽样的社区发现算法有很大提升；由表 4.4 可知，通过算法运行时间进行比较分析，在高效性方面本章提出的算法明显优于其他三种社区发现算法。

4.4 本章小结

本章通过将压缩技术和聚类方法应用于大规模复杂网络中，提出了一种基于压缩的大规模网络社区发现算法。该算法利用复杂网络中度分布的长尾特性，迭代地将网络中的边缘节点压缩到其它节点中，使网络的规模得以减少，之后在压缩后的网络中针对复杂网络的特性并利用划分聚类的思想完成了压缩网络的社区发现，最后完成之前被压缩过的节点的划分。通过与另外的算法进行比较分析，表明了该算法能够在保证准确性的同时提高计算效率。

第五章 总结与展望

5.1 总结

从上个世纪九十年代以来，针对复杂网络的研究已经成为许多领域研究中最重要的课题之一，比如社会学、计算机科学、生态学和经济学等诸多领域，并且已经成为上述诸多领域研究中最重要的课题之一。

本文首先对社区发现需要的基础知识进行了详细的介绍，包括图模型、度、平均度、度分布、最短路径长度、聚类系数和社区结构；之后介绍了现如今主流的算法原理和主要步骤并对比了它们的优缺点，并着重介绍了目前存在的适用于大规模网络的相关算法的效果与缺陷。通过梳理比较已有的算法技术，提出了下述两种社区发现算法：

基于抽样的大规模社区发现算法。该算法首先利用基于度的随机游走技术对整体网络进行抽样得到子图，然后将基于概要的发现算法应用到此子图上进行社区发现，之后获得初始社区，最后依据已有初始社区与未抽样的节点的相似度迭代式地将剩余节点进行划分。同时通过实验与其它社区发现算法进行了对比分析。

基于压缩的大规模社区发现算法。该算法首先根据复杂网络的长尾特性对复杂网络进行压缩折叠，利用划分聚类的思想选取初始社区中心，最后完成整个网络的社区划分。同时通过实验与其它社区发现算法进行了对比分析。

5.2 展望

作为复杂网络分析领域的一个重要课题，尤其是在大数据时代的背景下，社区结构的研究在很多领域都已引起足够重视，有着广泛的意义。本文虽然在大规模网络的社区发现方面做了一定的工作，但在此领域的研究仍然任重而道远，仍有很多的工作值得深入研究，我们今后的研究工作可以从以下几个方面着手进行：

（1）思考并实现将集成的思想应用到大规模社区发现算法中，通过将大规模网络划分成一些小块，对这些小块分别进行社区划分，最后通过集成方法将这些结果进行集成，最终实现高效且有效的社区划分。

（2）复杂网络中的社区结构具有层次性，即大社区中包含着小社区，如在大公司的组织架构中，不同的工作小组组成一个部门，不同的部门组成一个区域公司分支。在大规模网络中，这种层次性结构愈加明显。关于如何精确且高效识别大规模网络中层次结构的问题，值得进一步研究。

（3）大规模真实网络包含着大量的用户，随着时间的推移，很多用户的加入或

退出都会引起复杂网络的结构产生变化，使得不同时间段的社区发现结果差别较大。因此可以将研究社区结构的演化规律作为我们今后的研究重点。

参 考 文 献

[1] Chen J, Yuan B. Detecting functional modules in the yeast protein-protein interaction network[J]. Bioinformatics, 2006, 22(18):83-90.

[2] Dourisboure Y, Geraci F, Pellegrini M. Extraction and classification of dense communities in the web[C]. International Conference on World Wide Web. 2007:461-470.

[3] Roger G, Amaral L A N. Functional cartography of complex metabolic networks[J]. Nature, 2005, 433(7028):895-900.

[4] Boyd D. Why youth (heart) social network sites: The role of networked publics in teenage social life[J]. MacArthur Foundation Series on Digital Learning-Youth, Identity, and Digital Media Volume, 2007: 119-142.

[5] Krishnamurthy B, Wang J. On network-aware clustering of web clients[J]. ACM SIGCOMM Computer Communication Review, 2000, 30(4): 97-110.

[6] Wu F, Huberman B A. Finding communities in linear time: A physics approach[J]. Physics of Condensed Matter, 2004, 38(2):331-338.

[7] Jeyaratnarajah N. Cluster-Based Networks[J]. Ad Hoc Networking, 2001: 75-138.

[8] 张鑫, 刘秉权, 王晓龙. 复杂网络中社区发现方法的研究[J]. 计算机工程与应用, 2015, 51(24): 1-7.

[9] 王莉, 程学旗. 在线社会网络的动态社区发现及演化[J]. 计算机学报, 2015, 38(2): 219-237.

[10] Simon H A. The Architecture of Complexity[M]. Facets of Systems Science. Springer US, 1991.

[11] Weiss R S, Jacobson E. A method for the analysis of the structure of complex organizations[J]. American Sociological Review, 1955, 20(6): 661-668.

[12] Orman G K, Labatut V, Cherifi H. Qualitative Comparison of Community Detection Algorithms[J]. Communications in Computer & Information Science, 2011, 167(167):265-279.

[13] Fortunato S. Community detection in graphs[J]. Physics Reports, 2010, 486(3): 75-174.

[14] Girvan M, Newman M E J. Community structure in social and biological networks[J].

Proceedings of The Tational Academy of Sciences, 2002, 99(12): 7821-7826.

[15] 金弟, 刘大有, 杨博,等. 基于局部探测的快速复杂网络聚类算法[J]. 电子学报, 2011, 39(11):2540-2546.

[16] 金弟, 刘杰, 贾正雪,等. 基于 k 最近邻网络的数据聚类算法[J]. 模式识别与人工智能, 2010, 23(4):546-551.

[17] Shen H W, Cheng X Q, Guo J F. Quantifying and identifying the overlapping community structure in networks[J]. Journal of Statistical Mechanics Theory & Experiment, 2009, 7:07042.

[18] Kelley S, Goldberg M, Magdon-Ismail M, et al. Defining and Discovering Communities in Social Networks[M]. Handbook of Optimization in Complex Networks. Springer US, 2012: 139-168.

[19] Reid F, McDaid A, Hurley N. Partitioning Breaks Communities[M]. Mining Social Networks and Security Informatics. Springer Netherlands, 2013: 79-105.

[20] Xie J, 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.

[21] Nguyen N P, Dinh T N, Shen Y, et al. Dynamic social community detection and its applications[J]. PloS One, 2014, 9(4): e91431.

[22] Kernighan B W, Lin S. An efficient heuristic procedure for partitioning graphs[J]. Bell System Technical Journal, 1970, 49(2): 291-307.

[23] Barnes E R. An algorithm for partitioning the nodes of a graph[J]. SIAM Journal on Algebraic Discrete Methods, 1982, 3(4): 541-550.

[24] Newman M E J. Fast algorithm for detecting community structure in networks[J]. Physical Review E, 2004, 69(6): 066133-066133.

[25] Clauset A, Newman M E J, Moore C. Finding community structure in very large networks[J]. Physical Review E, 2004, 70(6): 264-277.

[26] Guimera R, Amaral L A N. Functional cartography of complex metabolic networks[J]. Nature, 2005, 433(7028): 895-900.

[27] Duch J, Arenas A. Community detection in complex networks using extremal optimization[J]. Physical Review E, 2005, 72(2): 986-1023.

[28] Agarwal G, Kempe D. Modularity-maximizing graph communities via mathematical programming[J]. The European Physical Journal B, 2008, 66(3): 409-418.

[29] Zhang Y, Wang J, Wang Y, et al. Parallel community detection on large networks with propinquity dynamics[C]. Knowledge Discovery and Data Mining. 2009:997-1006.

[30] Riedy J, Bader D A, Meyerhenke H. Scalable multi-threaded community detection in social networks[C]. In: Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International. IEEE, 2012: 1619-1628.

[31] Shi J, Xue W, Wang W, et al. Scalable community detection in massive social networks using MapReduce[J]. IBM Journal of Research and Development, 2013, 57(3/4): 12: 1-12: 14.

[32] Albert R, Jeong H, Barabási A L. Internet: Diameter of the world-wide web[J]. Nature, 1999, 401(6749): 130-131.

[33] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks[J]. Journal of Statistical Mechanics: Theory and Experiment, 2008, 2008(10): P10008.

[34] Gleich D F, Seshadhri C. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods[C]. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012: 597-605.

[35] 张新猛, 蒋盛益. 基于核心图增量聚类的复杂网络划分算法[J]. 自动化学报, 2013, 39(7): 1117-1125.

[36] Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters[J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): 2.

[37] Leskovec J, Kleinberg J, Faloutsos C. Graphs over time: Densification laws, shrinking diameters and possible explanations[C]. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. New York: ACM Press, 2005: 177-187.

[38] 柴变芳, 赵晓鹏, 贾彩燕, 等. 大规模网络的三角形模体社区发现模型[J]. 南京大学学报 (自然科学版), 2014, 50(4): 466-473.

[39] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.

[40] Arenas A, Fernandez A, Gomez S. Analysis of the structure of complex networks at different resolution levels[J]. New Journal of Physics, 2008, 10(5):4656-4658.

[41] Leskovec J, Lang K J, Dasgupta A, et al. Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters[J]. Internet Mathematics, 2008, 6(1):29-123.

[42] Richardson M, Agrawal R, Domingos P. Trust management for the semantic web[J]. Lecture Notes in Computer Science, 2003, 2870(10):351-368.

[43] Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters[J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1):2.

[44] Yang J, Leskovec J. Defining and evaluating network communities based on ground-truth[J]. Knowledge & Information Systems, 2012, 42(1):745-754.

[45] Leskovec J, Huttenlocher D, Kleinberg J. Signed networks in social media[J]. Proc Chi, 2010:1361-1370.

攻读学位期间取得的研究成果

[1] 梁晋，梁吉业，赵兴旺．一种面向大规模社会网络的社区发现算法[J]．南京大学学报(自然科学). 2016(01): 159-166.

致 谢

时光飞逝，转眼间三年的研究生生涯就要画上句点，这三年来有喜悦、有泪水、更有成长。所有的辛勤和奋斗都凝聚在了这一篇两万余字的毕业论文中，在此论文完成之际，特此向给予过我帮助和陪伴的人致以深深的感谢。

首先，我要向我的导师梁吉业教授和赵兴旺老师致以最衷心的感谢。梁老师是我在科研路上的启明灯和指路人，他用他渊博的知识、严谨的治学态度和忘我的工作精神使我尽快地进入了我的研究领域，并为我的学习和生活给予了诸多的建议与帮助。虽然梁老师会有很多繁重的行政事务需要处理，但他总是能在百忙之中抽出时间召开学术研讨会并会针对我在科研上所遇到的问题和瓶颈予以指导。赵老师既是一位良师益友，更是一位兄长，正是由于他一次次不厌其烦地指导我的算法思想和修改我的学术论文毕业论文，我才能取得现在的成绩。当我犯错的时候，他都不是批评我，而是更加细心地为我讲解我的困惑，让我学会了吸取教训和总结经验，同时，他优秀的科研能力值得我终身去学习。感谢这两位老师对我的悉心培养，教会了我许多道理，也让我顺利地完成了此篇硕士毕业论文。

感谢研究生班主任阎建红老师和冯旭东老师，多谢他们的关爱和照顾让我快速地成长；感谢本科班主任闫建霞老师，是她多年来的言传身教让我倍受鼓舞；感谢教学秘书张晓红老师在教学工作上的辛勤付出。

感谢所有同门的师兄师姐、师弟师妹给予我的帮助和关爱，感谢这个温馨团结、积极向上的团队，让我无论做什么事情都有着坚实的后盾和保障。

感谢宿舍的每一位舍友，武娟、张晶、王芳芳、王瑞花和苏娜，感谢我的同僚郭兰杰，感谢我的好朋友李璐、刘珏，正是他们的陪伴，让我能够愉快地面对学业的挑战与成长的痛。和他们相处的日子会成为我今生最美好的回忆，希望大家在毕业后仍然保持友谊，一起迎接更大的挑战。

感谢 2013 级研究生的所有同学们，三年时间让我们彼此了解，发生过的一切都仍历历在目，感谢大家对我的帮助。

最后我要感谢我的父母和家人，他们含辛茹苦地把我养大，让我有了健全的人格品格，是他们无私的爱、鼓励和支持，让我能够顺利完成学业，有了现在的成绩。参加工作之后，我一定会更加孝顺父母，常回家看看，将孝行得以践行。

在此毕业之际，太多太多的感谢和不舍萦绕心头，总之，在此我要对所有给予过我帮助和陪伴的人致以最为衷心的感谢，并祝福你们身体健康、事事顺利！

个人简况及联系方式

个人简况：

姓名：梁晋

性别：女

籍贯：陕西省洛川县

个人简历：

2009.09—2013.06 山西大学计算机与信息技术学院 本科

2013.09—2016.06 山西大学计算机与信息技术学院 硕士

联系方式：

E-mail：861924599@qq.com

承 诺 书

本人郑重声明：所呈交的学位论文，是在导师指导下独立完成的，学位论文的知识产权属于山西大学。如果今后以其他单位名义发表与在读期间学位论文相关的内容，将承担法律责任。除文中已经注明引用的文献资料外，本学位论文不包括任何其他个人或集体已经发表或撰写过的成果。

作者签名：

20　年　月　日

学位论文使用授权声明

本人完全了解山西大学有关保留、使用学位论文的规定，即：学校有权保留并向国家有关机关或机构送交论文的复印件和电子文档，允许论文被查阅和借阅，可以采用影印、缩印或扫描等手段保存、汇编学位论文。同意山西大学可以用不同方式在不同媒体上发表、传播论文的全部或部分内容。

保密的学位论文在解密后遵守此协议。

作者签名：

导师签名：

20　年　月　日