大规模社交网络的社区发现算法研究

朱杰

2018 年 5 月 30 日

大规模社交网络的社区发现算法研究

作　者　姓　名　朱杰

学　院　名　称　计算机学院

指　导　教　师　张欣

答辩委员会主席　***

申　请　学　位　工程硕士

学　科　专　业　软件工程

学位授予单位　北京理工大学

论文答辩日期　2018 年 5 月 30 日

An Algorithm for Large-scale Social Network Community Detection

Candidate Name:	Jie Zhu
School or Department:	Computer Science & Technology
Faculty Mentor:	Xin Zhang
Chair, Thesis Committee:	***
Degree Applied:	Master of Engineering
Major:	Software Engineering
Degree by:	Beijing Institute of Technology
The Date of Defence:	May 30，2018

大规模社交网络的社区发现算法研究

北京理工大学

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

作者签名：_____　　签字日期：_____

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：① 学校有权保管、并向有关部门送交学位论文的原件与复印件；② 学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③ 学校可允许学位论文被查阅或借阅；④ 学校可以学术交流为目的，复制赠送和交换学位论文；⑤ 学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

作者签名：_____　　导师签名：_____

签字日期：_____　　签字日期：_____

摘要

本文……。（摘要是一篇具有独立性和完整性的短文，应概括而扼要地反映出本论文的主要内容。包括研究目的、研究方法、研究结果和结论等，特别要突出研究结果和结论。中文摘要力求语言精炼准确，硕士学位论文摘要建议 500~800 字，博士学位论文建议 1000~1200 字。摘要中不可出现参考文献、图、表、化学结构式、非公知公用的符号和术语。英文摘要与中文摘要的内容应一致。）

关键词： 形状记忆；聚氨酯；织物；合成；应用（一般选 3 ～ 8 个单词或专业术语，且中英文关键词必须对应。）

Abstract

In order to exploit …….

Key Words: shape memory properties; polyurethane; textile; synthesis; application

目录

插图

表格

第 1 章　绪论

1.1　研究背景和意义

近年来，随着科技的发展和网络的不断普及，在线社交网络如今已经成为互联网时代最为基础的一部分。诸如微信、微博、Facebook、Twitter 和 GitHub 等等国内外的社交类平台软件的出现，使得人们可以更加有高效的沟通交流。在如今的移动互联网时代下，人们的社交重心由线下更多的转到了线上。线上的社交也确实带来了很多的便捷。人们不再有地域的限制，可以轻松地与亲朋好友时刻保持联系；人们通过社交平台可以迅速地认识了解一个人并与之成为朋友；人们可以扮演起在日常生活中无法扮演的角色，任何人都可以成为信息的分享者和传播者。

因此，在种类繁多的在线社交软件中，人们的参与度越来越高。新浪微博发布的 2018 年第一季度的财报显示，2018 年 3 月微博的月活跃用户数（MAUs）较上年同期净增约 7000 万，达到 4.11 亿，其中月活跃用户数中 93% 为移动端用户；2018 年 1 月 15 日在广州举行的微信公开课上，微信创始人、腾讯高级副总裁张小龙指出微信用户量已超 10 亿；而国外的 Facebook 更是早在 2017 年就已超 20 亿用户。社交网络具有传播迅速、传播广泛、自发性和言论相对自由等特点。除了普通个人用户，为了能够更快捷、更方便和更广泛的进行宣传，各类官方媒体也都已入驻这些知名在线平台。

面对着如此巨大的用户量，这些成熟的社交软件每天都会产生海量的用户数据，这些看似杂乱无章、毫无交集的数据中，其实蕴含了丰富的信息等待着人们去挖掘与分析。在这样的背景下，如果将社交平台中所有用户抽象成点，而用户与用户之间的关联抽象成边，这就抽象出了一张网络关系图，那么对于在互联网这个虚拟世界中形成的这一巨大而又复杂的社交网络的研究与分析就显得意义非凡。而面对如今的大数据环境，传统的一些研究分析手段都已经无法再胜任。因此，为了更好的利用社交网络给人们带来的便捷，同时又能避免产生危害，就产生了社交网络分析（Social Network Analysis）这一研究领域。它是一门横跨信息学、数学、计算机技术、社会学、管理学和心理学等学科的交叉科学，主要研究的是社交网络的网络结构和其演化、社交网络中的群体及其互动、社交网络中的信息及其传播。

正所谓“物以类聚，人以群分”。这道理也同样适用于社交网络。网络图内部连接比较紧密的节点子集合对应的子图被称之为社区（Community）。网络图中包含一个个

社区的现象称之为社区结构（Community Structure）。社区结构是社交网络结构的一个普遍特征。而给定一个网络图，找出其社区结构的过程就叫做社区发现（Community Detection）[1]。社区发现正是社交网络分析的一种手段。挖掘社交网络中的社区在人物分析、个性化商业推荐等领域有着很关键的作用。为了获得更多的用户群体、获取更多的流量关注或者刺激用户更多地进消费，在线求职招聘类平台分别要为求职者和招聘单位提供合适的岗位和应聘者；在线购物消费类平台要为用户推荐符合用户需求的商品；在线社交网络平台要为用户推荐兴趣相投的好友；在线新闻媒体平台要为用户推荐符合用户口味的相关讯息。在这样的背景下，一个优秀的个性化推荐系统就显得尤为重要。而在推荐系统中，免不了要对用户群体进行分类，而对社交网络的社区发现无非也就是对社交网络中的用户进行分类（或者说聚类）。在同一个社区内的用户群体，往往有着一些共性（比如相同的兴趣爱好）。对整个用户群体进行社区发现，那么在为某个用户进行商品个性化推荐时，可以重点推荐与该用户在同一个社区的其他用户已经购买过或者感兴趣的商品，这样用户对推荐商品的接受率会高很多。现在几乎所有的在线平台都可以绑定社交平台账号，在此前提下，对社交网络的社区发现对于商业推荐而言将会比以往更有意义。

随着用户量的越发增加，在线社交媒介的广泛使用带来的问题也越来越多，这也对社交网络的规范化和整治不断提出挑战。在人们享受社交网络带来的乐趣和便利之时，同样也有不法分子为了金钱或其他目的利用社交网络缺乏规范又利于传播等特点进行违法犯罪，包括诈骗和散布谣言等。近期国家广电总局也已经整治了一大批社交媒体，封杀了一系列严重违规的软件，即使是今日头条、抖音和快手这样的大公司也相继被勒令整改。对社交网络的社区发现研究同样可以对舆情监控这一领域做出一定的贡献，在锁定犯罪团伙之间的网络关系、挖掘舆情传播起源点、控制舆论导向等方面，社区发现算法均有一定的帮助。

此外，对大规模社交网络进行社区检测在社交网络用户数据的分布式存储上也有很大的意义。现在的社交网络数据量巨大，往往都是以图结构分布式存储于多台服务器。在图结构各个节点之间的通信开销之中，跨服务器的节点通信开销是主要的，而服务器内部通信往往比较快捷。因此，如果把隶属于同一社区的用户数据都存储于同一台服务器中，就可以很大程度上的降低服务器之间的通信开销，也就间接减少了各类图计算算法的运行时间。

[1]也有学者将”Community Detection” 翻译为社区检测、社团检测或者社团发现，本文将其称为社区发现，下文将统一采用“社区发现”。

对社区发现算法的研究其实远不止社交网络这一领域，社区发现也是对复杂网络的一种分析手段。除了社交网络，还有科学文献引用网络、生物学分子结构分析等等领域也都会使用社区发现算法。多年来众多学者的贡献使得对社区发现算法的研究已经比较成熟，但是依然存在着一些问题。其中最突出的就是：现如今的社交网络数据往往是海量的，对如此大规模数据的社交网络进行社区发现时，过往的一些经典算法显然已经无法胜任。因此，在这样的背景下，提出一种适应于当前大规模社交网络的社区发现算法就显得尤为迫切了。

1.2 国内外研究现状

社区发现算法根据不同的角度可以有很多种分类，本文将现有的社区发现算法分为三大类：基于链接的方法，基于内容的方法和融合链接与内容的方法。基于链接的方法也就是基于网络拓扑结构的方法，它将社交网络看做用户为节点、关系为边的网络图；基于内容的方法也被称为基于主题的方法或者基于节点相似度的方法，主要是基于社交网络中用户的个人信息以及发表过的内容来进行社区发现。而融合的方法则同时关注网络拓扑结构和用户属性，以此来获取更高质量的社区。

McPherson 等人最早提出了同质性原则：相似性产生联系[1]。这也被认为是社区定义最早的参考准则，同时这也有利于发现社区。人们总是习惯于和自己相似的人结交关系，不论是友谊还是工作关系，任何关系网络中人们的缔结方式均有此趋势。因此，在社交网络中两个用户之间所谓的链接关系被认为是一个可以证明彼此之间是有着某些共同点的证据。

在基于链接的社区发现算法中，社交网络是由图为模型，节点代表社区成员，边代表成员之间的关系或者交互。这里社区所需的凝聚力属性就是成员之间的链接。在社区之中链接较为密集，而在社区之间链接相对较为稀疏。文献[2] 分别将原始图结构中的组件和派系当做是已知的社区。然而，更多更有意义的社区需要通过基于图划分（聚类）的方法来检测得到，其目标就是尽可能减少社区之间边的数量，并且使得同一个社区中节点之间有更多的内连接，而与别的社区中节点的外连接就可以减少。大部分的方法都是基于二分迭代，即不断地将一个社团划分为两个社团。然而在复杂网络中社区的数量显然是无法预先得知的，在这个层面上，GN 算法[3] 是最为广泛使用的基于链接的社区发现算法。GN 算法的基本思想是删除社区之间的连接，使得剩下的每个连通部分就是一个社区。作者巧妙地借助了最短路径这一思想。GN 算法中定

义一条边的边介数（Betweeness）为网络中所有节点之间的最短路径中通过这条边的数量，而边介数高的边要比介数低的边更可能是社区之间的边。因为两个社区中的节点之间的最短路径都要经过那些社区之间的边，所以它们的边介数会很高。

然而，GN 算法的一个缺点就是无法评价找出的社区发现结果是否最优。为了解决这一问题，就必须有一种度量的方法，可以在计算过程中来衡量每个结果是不是相对最佳的结果。这同样也是算法好坏的评价指标。Newman 等人提出了模块度（Modularity）[4] 这一概念。模块度的大小定义为社区内部的总边数和网络中总边数的比例减去一个期望值，该期望值是将网络设定为随机网络时同样的社区分配形成的社区内部的总边数和网络中总边数的比例的大小，模块度一般记为 Q。在每次划分的时候计算 Q 值，当 Q 取最大值时则是此网络较为理想的划分。Q 取值，越大越好，实际中一般 Q 最高点在 0.3 至 0.7。有时候，当不能或者不容易获取全部网络的数据时，可以用局部社区中的局部模块度来检测社区的合理性。局部模块度比全局模块度快很多，中小网络效果会比全局的差些，但是中等或大规模的网络中，局部模块度效果可能好要比全局的更好。GN 算法还有一个缺点就是时间复杂度过高，只适用于规模较小的网络。对于数据量惊人的社交网络，GN 算法约 $O(N^3)$ 的时间复杂度明显无法胜任对其社区发现的工作。

考虑到社交网络的数据集过于庞大，要想适应于大规模的社交网络，算法的速度就显得至关重要。而这就必须要提到 Raghavan 等人提出的 LPA 算法[5] 了，这是迄今为止速度最快的社区发现算法，仅仅只有线性时间的复杂度，简单易理解且不需要任何先验信息。但是该算法由于标签更新阶段是随机顺序，并没有考虑不同节点的重要性，容易导致结果产生震荡，实验结果通常不稳定。研究人员围绕 LPA 算法不断进行改进，在算法的传播规则、收敛条件和更新策略等方面进行相关研究。例如朱福喜等人提出了一种基于标签传播概率的算法[6]。

基于链接的社区发现算法其实也可以被看作是一种数据挖掘或者说机器学习聚类算法，相当于无监督的用户分类。因此这其中可以用到的无监督学习的相关技术包括：k-means 算法、混合模型和层次聚类等等。此外，其他的一些可应用于社区发现的分图算法还包括：最大流最小割理论[7]、谱二分的方法[8]、Kernighan-Lin 划分算法[9] 和最小电导率分割算法[10] 等等。

尽管基于链接的技术更加直观且基于社会学的同质原则，但也有两个原因导致它们在识别基于相似兴趣的用户社区方面存在缺陷。首先，许多社交关系不是基于用户

的兴趣相似性，而是基于其他因素，如朋友和亲属关系，并不一定反映用户间的兴趣相似性。其次，许多有着相似兴趣的用户彼此之间可能并没有互相关注，以至于在网络之中似乎是没有关联的[11]。随着在线社交网络功能的不断增加，网络上除了用户之间的链接之外，还有许多用户自己提交的内容（称为社交内容）可用。用户可以维护个人资料页面，撰写评论，分享文章，标记照片和视频以及发布他们的状态更新。因此，研究人员已经探索了利用社交内容的主题相似性来检测社区的可能性。他们提出了基于内容或主题的社区检测方法，这样一来，不论社交网络结构如何，都可以检测到志同道合的社区用户[12]。

大多数基于内容的社区发现工作都侧重于检测社区文本内容的相似模型。比如，Abdelbary 等人提出的算法[13] 利用了高斯受限玻尔兹曼机（GRBM）来识别主题社区。尹志军等人[14] 将社区发现与主题建模结合在一个统一的生成模型中，以检测在结构关系和潜在主题方面相互一致的用户群体。在他们的框架中，一个社区可以围绕多个主题形成，一个主题也可以在多个社区之间共享。Sachan 等人[15] 提出了概率方案，将用户的帖子、社交关系和交互类型结合起来发现 Twitter 中的潜在用户社区。在他们的工作中，他们考虑了三种类型的互动：传统推文、回复推文和转载推文。其他学者还提出了隐含狄利克雷分布模型（LDA）的变体来识别主题社区，例如作者-主题模型（Author-Topic model)[16] 和社区-用户-主题模型（Community-User-Topic model)[17]。

另一个流派的工作将基于内容的社区发现问题建模为图聚类问题。这些方法都基于相似性度量标准，该度量标准能够根据用户都感兴趣的主题计算用户的相似度，并基于聚类算法来提取具有相似兴趣的用户群体（潜在社区）。例如，刘洪涛等人[18] 提出了一种基于用户间主题距离（Topic-Distance）的聚类算法来检测社交标签网络中基于内容的社区。在这项工作中，LDA 被用来提取标签中隐藏的主题。彭敦陆等人[19] 提出了一个层次聚类算法来检测推文中的潜在社区。他们在新浪微博中使用了预定义的类别，并根据用户在每个类别中的兴趣程度计算了用户的配对相似度。

像基于链接的方法一样，基于内容的社区发现方法也可以转化为数据的聚类，这里的一个社区只是一组节点的集合。代表用户的节点与同一社区内的节点相似度较高，而与社区外的节点相似度较低。从这个意义上说，亲密关系确实是社区所需的凝聚力属性。

基于内容的方法其实是为常规文本设计的，但是诸如 Twitter 或微博这类社交网络多是简短、混杂和非正式的社交内容。在这种情况下，社交内容本身并不是提取真

实社区的可靠信息[20]。通过社交结构（即链接）丰富社交内容有助于我们找到更有意义的社区。研究人员们已经提出了几种方法将链接和内容信息结合起来用于社区发现。正如参考文献[21,22]中所述，它们可以拥有更好的性能。大多数这类方法是通过共享隐含变量这一手段来为社区成员制定链接和内容的综合生成模型。

社区发现算法的常见方法是将网络划分为不相交的社区成员，而这种方法忽略了个体可能属于两个或更多社区的可能性。但是，许多真实的社交网络都存在着社区的重叠[23]。例如，一个人可以属于多个社交群体，例如家庭群体和朋友群体。越来越多的研究人员开始探索允许社区重叠的新方法，即重叠社区（Overlapping Communities）。重叠社区引入了另一个变量，即不同社区中用户的成员身份，称为 cover。由于与标准社区相比，重叠社区有大量可能的 cover，因此检测此类社区代价就很高。

一些重叠的社区发现算法利用网络中用户的结构信息将网络的用户分成不同的社区。这类方法的主导算法是基于集团渗透理论（Clique Percolation Theory）[24]。然而，LFM 和 OCG 方法是基于对用户出入度适应函数的局部优化[25,26]。此外，一些模糊社区发现算法会计算每个节点属于每个社区的可能性，如 SSDE 和 IBFO[27,28]。几乎所有的算法都需要先验信息来检测重叠的社区。例如，LFM 需要一个参数来控制社区的大小。不过也有一些基于相似性的方法将社区看作分布在整个用户空间的隐含变量，如参考文献[29]。

Erosheva 等人[30] 介绍了 Link-LDA，一种重叠的社区发现方法。它可以同时根据摘要（内容）和参考文献（链接）对科学类论文进行分类。在它们的生成模型中，论文被假定为摘要和参考文献的一对模型，每个部分都用 LDA 抽取特征。在摘要和参考文献中相似性都很高的文章倾向于有着相同的主题。与 Link-LDA 相反，Nallapti 等人[31] 没有将参考文献视作待处理的单词，并提出需要明确引用文本和参考文献之间的主题关系。他们提出了 Pairwise-Link-LDA 来模拟文档对之间的链接存在，并通过使用这些附加信息获得了更好的主题质量。其他利用 LDA 融合链接和内容的方法可以参考文献[32,33]。除了相似度生成模型外，还有其他一些方法将链接和内容信息结合起来用于社区发现，如谱聚类中利用矩阵分解和核聚变的方法[34,35]。

1.3 论文主要工作

通过查阅大量关于社交网络、社区发现、聚类等方面的文献资料，深入理解社交网络及重叠社区特性的基础上，认真研究社区发现相关算法，本文完成了如下工作：

（1）CDABSLP 算法的设计。该算法从两个方面改善 LPA 算法不稳定的问题：首先，计算网络中每个节点的节点影响值并按节点影响值降序排列作为节点标签更新的顺序，取代传统 LPA 算法中节点更新顺序随机确定的方法；其次，在每次标签更新迭代过程中，当传统的标签计算方法返回多个标签时，提出一种新的标签计算公式，计算返回标签的影响强度，在返回的多个标签中重新选择一个影响强度最大的标签作为该节点的新标签，以此替代传统 LPA 算法中随机选择一个标签的方法。CDABSLP 算法既保持了传统 LPA 算法的优点，还解决了 LPA 算法不稳定的问题，该算法能够得到稳定的社区结构。大量实验结果表明 CDABSLP 算法的性能优于目前一些代表性的社区发现算法。

（2）OCDABSLP 算法的设计。该算法 OCDABSLP 算法采用同步更新策略，在标签更新过程中，当一个节点拥有的所有标签对应的隶属度都小于 $1/v$，且此时有多个标签的隶属度同时取最大值时，将节点影响值引入到标签隶属度计算公式中，得到这些标签的影响强度，保留影响强度最大的标签，取代传统 COPRA 算法随机保留其中一个标签的方法，提高算法的稳定性。在重叠 LFR 数据集上的实验结果表明 OCDABSLP 算法解决了 COPRA 算法不稳定的问题，能够检测得到较优的重叠社区结构，验证了本章提出的稳定策略在重叠社区发现算法 COPRA 算法中的适用性。

1.4 论文组织结构

本论文主要对重叠社区的挖掘算法进行研究，并设计了相应的优化算法。本文主要包括四大章节，其主要的结构组织如下：

第一章为绪论。主要介绍了课题的背景、意义、国内外现状以及本课题的主要研究内容。其中，重点介绍了各类社区发现算法的国内外研究现状。

第二章为相关工作。首先介绍了复杂网络中社交网络的相关概念，包括社交网络概述、社交网络的统计特性以及社交网络的典型特征；接下来从不同角度给出了社区的定义，从而引出了社区发现这一概念。

第三章为基于稳定标签传播的非重叠社区发现算法。主要介绍本文设计的一种基于节点影响度的稳定的标签传播非重叠社区发现算法（CDABSLP）。本章首先是对经典标签传播算法做了简单介绍；然后详细介绍 CDABSLP 算法的设计思路、核心思想和关键步骤等，接着在真实网络以及人工基准网络上的实验，并与其他基准算法进行对比实验，以此来分析算法的效果，最后进行实验总结。

第四章为基于稳定标签传播的重叠社区发现算法。主要介绍本文设计的一种基于节点影响度的稳定的标签传播重叠社区发现算法（OCDABSLP）。本章先简单介绍了多标签传播算法（COPRA），然后详细介绍 OCDABSLP 算法的设计思路、核心思想和关键步骤等，接着在真实网络以及人工基准网络上的实验，并与其他基准算法进行对比实验，以此来分析算法的效果，最后进行实验总结。

第 2 章　相关工作

2.1　社交网络

2.1.1　社交网络概述

在维基百科中，社交网络（Social Network）被定义为“由许多节点以及节点间关系构成的一个网络结构。节点通常是指个人或组织（又称社团），社交网络代表各种社会关系”。对社交网络的分析在早期只是针对现实生活中切实的方便调查的关系进行分析。比方说：早期在国外曾有研究人员在研究如何减少政府机构的冗余行政人员以提高办事效率和降低政府开销时，就使用到了社交网络分析这一手段。他们采用私下采访和调查的手段获取了某一政府机关几乎全部工作人员之间的来往接触关系，建立了一张交际网络。通过对这张交际网络的分析发现，其中有些节点在业务流程线上是属于多余的，其功能只是交接两边的节点。对于提高效率而言，分析此网络并减少这样无谓的节点即可有效的降低开销。不像早期的社交网络主要是通过合作关系建立起来的职业网络，如今随着互联网社交媒体的诞生和飞速发展，社交网络逐渐线上化。

本文所指的社交网络特指在线社交网络（下文统称社交网络）。现在人们所说的社交网络一般而言其实就指代在线社交软件，也就是在互联网上与其他人产生联系的一个平台。在线社交媒介主要有即时通讯类软件（比如微信、QQ）、在线社交类软件（比如 Facebook、人人网）、微博类软件（比如新浪微博、Twitter）、贴吧类软件（比如百度贴吧、悟空问答、知乎）、博客分享类软件（比如 CSDN、简书）、职场关系类软件（比如领英、脉脉）和短视频分享类软件（比如抖音、快手）等等。而社交网络就是在这些社交媒介中抽象虚拟出来的一张网络图，在这张图中，每个个人或者组织抽象为一个节点，而人与人之间的关系或者互动则抽象为边。每个账号在社交媒体上填写的个人信息就是其节点属性，同样节点彼此之间的边上也有着相应的边的属性。这一切就构成了一个社交网络结构。网络虽然都是抽象出来的，但是这些关系却又都是真实的。图2.1展示了一个简单的社交网络抽象图。

图 2.1　一个简单的社交网络抽象图

2.1.2　社交网络的统计特性

因为社交网络的本质其实就是一个由节点（人或组织）和边（社会关系）组成的图结构，所以说社交网络模型之中的很多概念都是来源于图论。在这一小节中，将会简单介绍社交网络中常用的几种统计概念，包括节点的度及其分布、网络密度、平均路径长度、边的介数和聚类系数。这些统计概念或多或少都旨在反映社交网络的一些特性，比如疏密程度、信息传播开销等等。

（1）节点的度（Degree）。在无向图中，任意节点的度即是与其相连的边的数目。而在有向图中，又可以细分为入度和出度。任意节点的入度就是以该节点为终点的边的数目；同样的，任意节点的出度就是以该节点为起点的边的数目。在社交网络之中，一个节点的度越大，就表示其在这个网络中扮演着越重要的角色。影响力越大的人，在网络中节点的度就越大。比如说在微博上，拥有众多粉丝的明星们，他们在社交网络中抽象出的节点度就很大，而普通用户往往只有很少的粉丝，其度就很小。网络的平均节点度就是网络中所有节点的度的平均值，它可以反映网络的疏密程度。此外，还可以通过节点度的分布来刻画描述不同节点的重要性。

（2）网络密度（Density）。在社交网络之中，网络密度被定义为网络中实际存在的边数与最多可容纳边数的比值。通常被用来测量社交网络中社交关系的紧密程度及其演变趋势，其计算方式详见公式2.1。如果一个社交网络的网络密度还很小，则说明

该网络还尚且处在起步阶段；而若一个社交网络的网络密度已经比较大了，那么说明该网络已经比较成熟，网络之中几乎所有节点之间都有联系。

$$Density = \frac{2m}{n(n-1))} \tag{2.1}$$

公式2.1中的 n 和 m 分别为社交网络中边的数目和节点的数目，且 $Density \in [0,1]$。其中，当整个网络中没有一条边，即所有节点都独立存在时，$Density$ 取 0；而当网络中所有节点之间都有边相连时，即网络处于全连接状态时，$Density$ 取 1。一般而言，大规模的社交网络的密度会比中小规模的小一些，因此，不同规模之间的网络也就不具有可比性了。这也不难理解，举个简单的例子，以学校为规模建立一个社交网络和以一个家庭为规模建立一个社交网络，显然以一个家庭社交网络的网络密度会大很多。

（3）平均路径长度（Average Path Length）。一个社交网络的平均路径长度被定义为任意两个节点之间的最短路径的平均长度，也就是任意两个节点之间的最短关系路径上节点个数的平均值。其计算方法详见公式2.2。

$$APL = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \tag{2.2}$$

公式2.2中的 n 为当前网络节点个数，d_{ij} 为网络中节点 v_i 和 v_j 之间的最短路径。平均路径长度 APL 通常也是用于反映网络间的紧密程度的，一般也可叫做网络的平均距离。如果社交网络的平均路径长度比较大，则代表网络比较稀疏，节点之间进行信息传播的开销比较大；相反，若社交网络的平均路径长度小，则代表网络稠密，节点之间可以比较迅速快捷的进行传递消息。

（4）聚类系数（Clustering Coefficient）。根据图论，聚类系数表示的是一个图中节点汇聚程度的系数。在很多社交网络中，若节点 v_i 与节点 v_j 相连接，而节点 v_j 与节点 v_k 相连接, 那么很大概率上节点 v_i 和节点 v_k 也会相连。这种现象也表明了社交网络中部分节点之间存在着密集连接的这一特性。在无向图中，节点 v_j 的聚类系数 CC_{v_j} 的计算方式详见公式2.3。

$$CC_{v_j} = \frac{n}{C_k^2} = \frac{2n}{k(k-1)} \tag{2.3}$$

公式2.3中 k 表示节点 v_j 所拥有的邻居节点数目，n 表示节点 v_j 的所有相邻节点

之间互相连接的边的数目。简单来说，聚类系数可以用来描绘社交网络中一个用户朋友们之间也是朋友的概率，反映的也就是社交网络的聚集性。具体的，它还可以分为全局聚类系数和局部聚类系数，这里不再赘述。

（5）介数（Betweeness）。介数可以分为节点介数和边介数，表示的是网络图中某一节点或者某一条边被整个图中所有节点间的最短路径经过的概率之和。通常是用来评价节点的重要程度的。比方说在连接不同社区之间的中间节点（或者边）的介数就会比其他节点（或者边）的介数要大很多，这也反映了这类节点在社交网站中作为消息传播的核心地位及其重要程度。对于网络中任意节点 v，其介数的计算方式详见公式2.4。

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.4}$$

在公式2.4中，$\sigma_{st}(v)$ 表示经过节点 v 的 $s \to t$ 的最短路径条数，σ_{st} 表示 $s \to t$ 的最短路径条数。直观上来看，节点 v 的介数 $C_B(v)$ 反映的是节点 v 作为“桥梁”或者“枢纽”的重要程度。

2.1.3 社交网络的典型特征

在社交网络之中普遍存在着两个典型的特征：小世界效应和无标度特性。

（1）小世界效应（Small-world Effect）。在 1929 年匈牙利作家 F.Karinthy 率先提出了“小世界现象”的论断。他认为，地球上的任何两个人都可以平均通过一条由 5 位联系人组成的链条而联系起来。在 1967 年，美国哈佛大学的社会心理学教授 Stanley Milgram 提出了著名的“六度分隔（Six Degrees of Separation）假说”，大意同样为任何两个想要取得联系的陌生人之间最多只隔着 5 个人，便可完成两人之间的联系。他通过设计了一个信件实验来证明他的猜想，实验大致经过为：他随机选择了 300 多人，每人分发了一封信并指定了各不相同的收信人；要求如果寄信人认识收信人，则直接寄出，否则就寄给一个自己认识的并且可能认识收信人的人，直至收信人收到信为止；实验最终共有约 60 人收到了信，而这些信平均经手了 6 次就到达了收信人手中。在 1998 年的时候，Duncan Watts 和 Steven Strogatz 正式提出了小世界网络的概念并建立了小世界模型[36]。文中将小世界效应定义为：若网络中任意两个节点之间的平均距离（即平均路径长度 APL）随网络中节点数 n 的增加呈对数增长，即 $APL \sim ln(n)$，

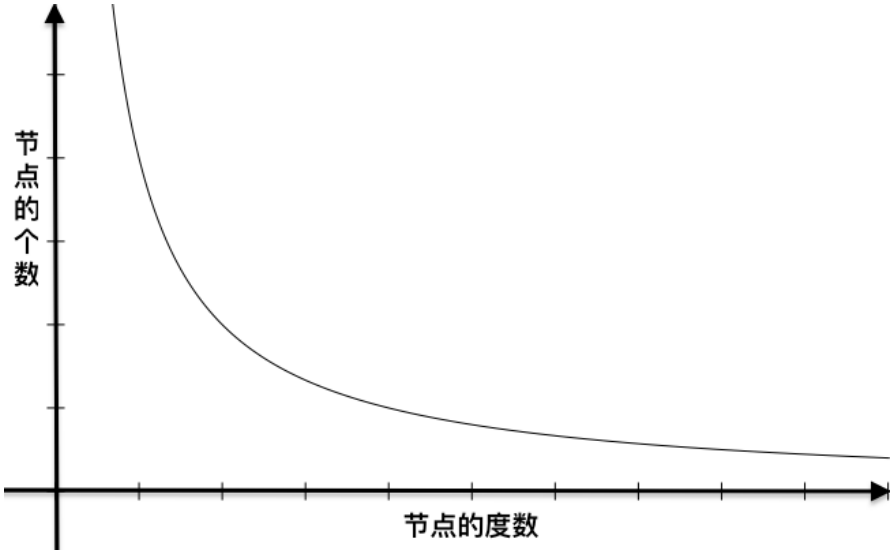

图 2.2　社交网络中度的幂律分布曲线

且网络的局部结构上仍然具有较明显的集团化特征。

小世界效应反映的是社交网络中任何用户之间都近在咫尺的现象，简单来说就是社交网络的平均路径长度都很短。小世界现象在在线社交网络中得到了很好地验证，根据 2011 年 Facebook 数据分析小组的报告，Facebook 约 7.2 亿用户中任意两个用户间的平均路径长度仅为 4.74，而这一指标在 Twitter 中为 4.67。因此可以说，在五步之内，任何两个网络上的个体都可以互相连接。

（2）无标度特性。大多数社交网络都存在着少数节点的度极大，而大部分节点都只有较小的度这一现象。其网络缺乏一个统一的衡量尺度而呈现出异质性，我们将这种节点度分布不存在有限衡量分布范围的性质称为无标度。这其实体现的是社交网络中用户的度呈现出幂律分布的规律。其实幂律分布广泛存在于各个领域，其核心就是绝大部分事件的规模其实很小，但是极少数事件的规模却表现的相当大，直观上就像图**??**中幂函数曲线一样。举几个例子，比如说世界上绝大部分的财富都被掌握在极少数的超级富豪们的手中；再以网站的访问量来说，尽管互联网之上为广大网民提供了无数的网页，但是每天大家访问量最多的也就是那么几个熟悉的网页；又比方说在微博上，一个明星的粉丝可能有上百万上千万，但是大部分人也只有寥寥无几的粉丝关注量。幂律分布其实体现的是一种极端的不平衡性。

2.2 社区发现

2.2.1 社区的定义

社交网络除了小世界效应和无标度特性这两个典型特征之外，还有一个重要的一个特点就是“社区（Community）”的存在。也有一些学者将之译为“社团”，本文将之称为社区。在直观上，同现实生活中所说的社区一样，社交网络乃至复杂网络中的社区也可以简单地被理解为是一些彼此之间联系紧密的节点的集合，或者说是一些拥有共同或相似属性的个体组成的团体，这些集合或者团体对于分析社交网络有着至关重要的意义。目前在学术界对社区的概念尚且还没有一个完整统一的定义。本文下面从不同的角度给出了三种定义：基于网络拓扑结构的定义、基于节点相似度的定义和重叠社区的定义。

（1）基于网络拓扑结构的社区定义。将社交网络抽象成一个由节点（人或组织）和边（社会关系）组成的网络拓扑结构，那么一个社区就是指由若干个节点组成的具有高内聚特性的子集合。在这个子集合之中的节点彼此之间联系紧密，表现上就是存在着相对比较多的边；而在多个子集合之间，也就是社区之间联系比较稀疏，表现上就是存在着相对较少的边。图2.3所示即是一个具有社区结构的网络。图中的 14 个节点组成的网络拓扑图中形成了 3 个社区结构，为了更直观显示，三个社区的节点分别以黄色、蓝色和绿色三种颜色区分。此种定义对应的社区发现算法即是本文绪论第二小节国内外研究现状中提到的基于链接的社区发现算法。

（2）基于节点相似度的社区定义。这里定义的社区依然是由若干节点组成的子集合，但是在表现形式以及定量分析手段上，这与基于网络拓扑结构的定义不同。该定义假定一个社区内部的节点都是相似的，而社区之间的节点的相似度则较低。节点之间的相似度的高低依靠建立节点相似度模型来衡量。这在直观上也不难理解，因为社区本就是代表着社交网络中具有相同或者类似属性的元素的子集。此种定义对应的社区发现算法即是本文绪论第二小节国内外研究现状中提到的基于内容的社区发现算法。

其实在本质上，基于网络拓扑结构的社区定义和基于节点相似度的社区定义是一致的。两者都是将社区定义为网络中所有元素组成的集合的若干子集，在子集之中的元素基于某种因素会彼此连接紧密，而与其他子集内的元素连接稀疏。只不过两者对于元素之间的紧密与稀疏的定量分析手段不同，前者是根据网络的拓扑结构，根据节

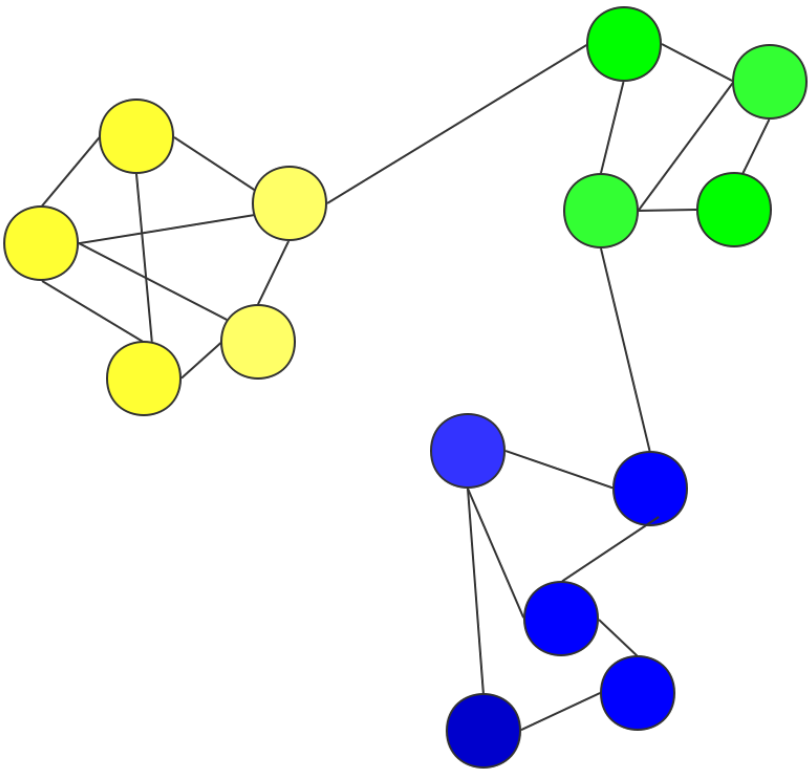

图 2.3　一个具有社区结构的网络示意图

点之间的边的联系，而后者是根据节点的属性分析节点间相似度。

（3）重叠社区的定义。上文关于社区的定义都忽视了个体可能属于两个或更多社区的可能性。但是，许多真实的社交网络都存在着社区的重叠。例如，一个人可以属于多个社交群体，例如家庭群体和朋友群体。图2.4所示即是一个具有重叠社区结构的网络。图中的 16 个节点组成的网络拓扑图中形成了 3 个社区结构，三个社区的节点分别以黄色、蓝色和绿色三种颜色区分，而这其中有两个黄蓝相间的节点即是同时属于黄色代表的社区和蓝色代表的社区。重叠社区（Overlapping Communities）和普通社区相比，唯一的差别就在于，代表重叠社区的子集之间可以有交集，而代表普通社区的子集之间是没有交集的。而在包含重叠社区的网络中，这些重叠社区间的交集，也就是这些属于多个社区的元素（个体），对社区的演化和社区间的沟通都起到了极其重要的作用，因为它们就是不同社区之间的桥梁和纽带。一般而言，重叠社区结构也可以分为两种。一种是离散型重叠社区，即一个节点要么属于某一个社区，要么不属于这一个社区。另一种是模糊型重叠社区，即每个节点有着对于不同社区的隶属度。关于重叠社区的定义对应的社区发现算法即是本文绪论第二小节国内外研究现

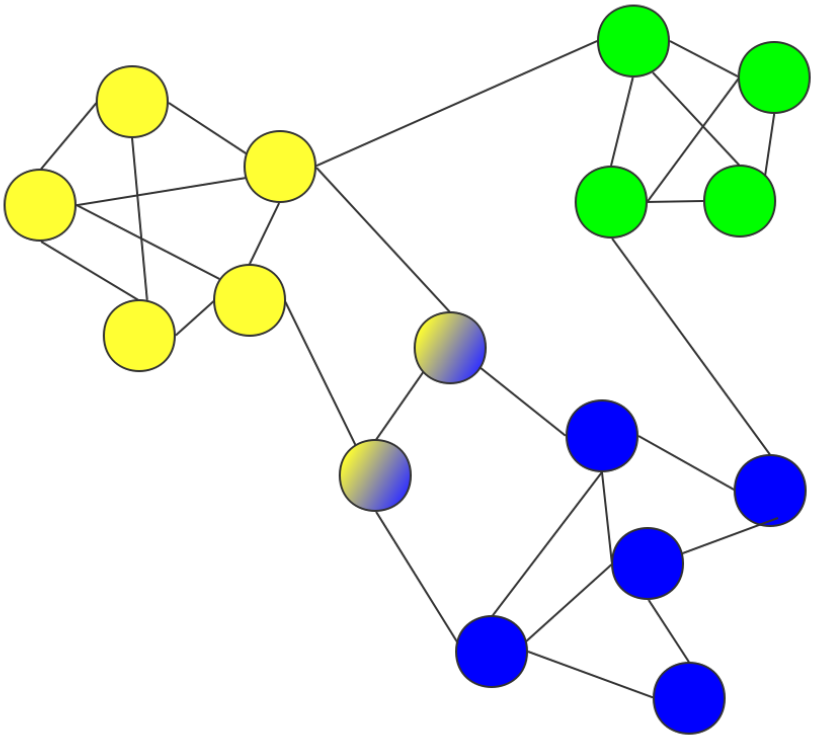

图 2.4　一个具有重叠社区的网络示意图

状中提到的重叠社区发现算法。

2.2.2　社区发现概述

上一小节已经较为详细的解释了社区的含义。那么给定一个网络图 $G = (V, E)$，其中顶点集为 V，边集为 E，找出其社区结构的过程就叫做社区发现（Community Detection，也可以译作社区检测）。社区发现是一个复杂而有意义的过程，它对研究社交网络乃至复杂网络的特性具有重要作用。挖掘社交网络中的社区在人物分析、商业个性化推荐和舆情控制等领域有着很关键的作用。

在社交网络之中，社区结构是客观存在的，但某个社区内的某个用户只和那些与其有直接边相连的用户产生互动，殊不知在这个社区内，他和那些与其没有直接边相连的用户其实也很“近”，如果要做好友推荐，属于同一社区的用户之间就应该优先进行推荐。此外，“物以类聚，人以群分”，对一个大型网络调用社区发现算法，其实是对其按照某种标准进行了划分，在此基础上可对每个社区做进一步的发掘。而从计算的角度而言，社区划分相当于分解了任务，起到了降低计算复杂度的作用。并且目前

的社交软件上用户广泛，不可能将所有用户的信息都存储在同一台服务器之上，这就必须要使用到分布式存储。在分布式存储中，如果不经过分类杂乱无章的进行存储，那么最终将导致巨大的通信开销，而如若先将社交网络进行社区发现，把同一社区内的用户存储在同一服务器内，这就可以省下大笔的通信开销，大大提高社交平台的性能。

近几年，发现及分析社交网络中的社区结构得到了许多学者的关注，同时也出现了很多的社区发现算法，在本文绪论第二小节国内外研究现状中将社区发现算法分为了基于链接的算法、基于内容的算法和融合了链接与内容的算法。在针对社区是否重叠上，还可以将其分为非重叠社区发现和重叠社区发现。对于具体的算法，此处不再赘述。

第 3 章　基于稳定标签传播的非重叠社区发现算法

目前已经有大量的社区发现算法被提出，但是针对社交网络小世界、无标度的特性以及面对大规模的数据量，对算法的时间复杂度提出了严格的要求。而在第一章节国内外研究现状中也曾提到，标签传播算法（LPA 算法）是至今为止执行最快的社区发现算法之一。LPA 算法具有接近线性的时间复杂度，算法设计简单并且无需参数，引起了国内外学者的关注。然而，LPA 算法也存在一些不足，如社区划分结果不稳定并且鲁棒性差等。针对 LPA 算法的不足，一些改进算法相继被提出出。例如 Leung 等人[37] 在传统 LPA 算法中加入启发式思想，引入标签在节点的 hop score 值，改进 LPA 算法的效率和性能。文献[38] 通过改进 LPA 算法的迭代结束条件，提高 LPA 算法的效率。LabelRank 算法[39] 引入标签概率矩阵，消除了 LPA 算法的随机性，避免了多次输出的不一致问题。LPA 算法是最早的基于标签传播的非重叠社区发现算法。

本章将提出一种基于稳定标签传播的非重叠社区发现算法（Community Detection Algorithm Based on Stable Label Propagation）, 下文简称 CDABSLP。CDABSLP 算法通过固定标签更新过程中节点的顺序，并且改进节点标签更新时标签选择的方法来提高 LPA 算法的性能。首先，算法计算网络中每个节点的影响值作为节点重要性的评判指标，并按照节点影响值降序排列作为标签更新过程中节点的顺序；然后，算法迭代的执行标签传播过程，直到检测到网络的社区结构。在每次标签传播过程中，CDABSLP 将节点影响值引入到标签计算公式中构造新的标签计算方法，计算邻接点中出现的每一个标签的重要性，更新节点标签。满足终止条件后，算法根据节点的标签将其划分到相应社区中，得到最终的社区划分结果。

本章接下来的内容组织结构上将先简单介绍下标签传播算法，然后详细介绍 CDABSLP 算法的设计思路、核心思想和关键步骤等，最后在真实网络以及人工基准网络上的实验，并与其他基准算法进行对比实验，以此来分析算法的效果。

3.1　标签传播算法 (LPA)

2007 年，Raghavan 等人[5] 首次将标签传播算法（LPA）应用到复杂网络社区发现中，LPA 算法的主要思想是利用网络的拓扑结构引导算法检测网络的社区结构。初始的时候，LPA 算法为每个节点分配一个唯一的各不相同的标签，表示开始的时候所有

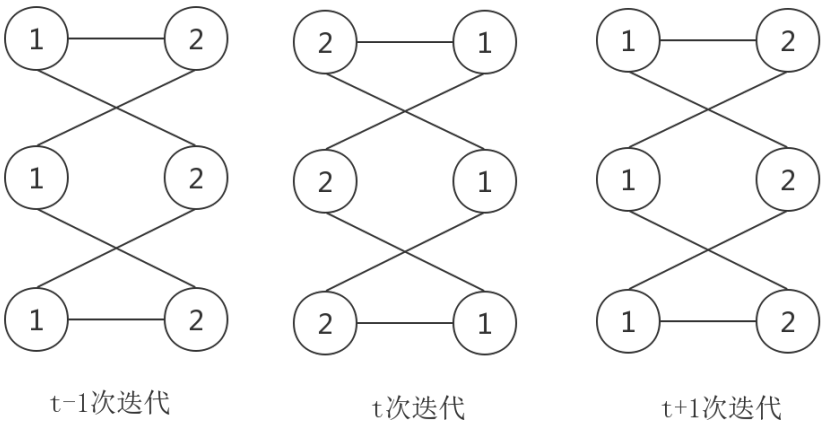

图 3.1　二分网络中的标签震荡现象

节点都各自组成一个社区。然后重复的进行标签更新过程，每执行一次，每个节点将自身的标签更新为它的邻居节点中最普遍存在的标签。如果多个标签在它的邻接点中出现的频次最高，那么 LPA 算法将随机的选择其中的一个标签赋给该节点。在这个重复的过程中，联系密集的节点逐渐将它们的标签更新为相同的标签，最后 LPA 算法根据节点的标签将其划分到相应社区中。

公式3.1为 LPA 算法节点标签更新的公式。

$$c_i = \arg\max_l \left|\Gamma_i^l\right| \tag{3.1}$$

其中 Γ_i^l 表示标签 l 的节点 i 的邻接集合。

LPA 算法思想简单，容易理解，但是，在多次迭代后，算法并不能保证收敛。在二分网络或近似二分网络中，当算法采用同步标签更新策略时，每个节点根据其邻接点在上一次更新后得到的标签计算它本身新的标签，此时可能出现标签震荡现象。如图3.1所示，二分网络中节点的标签在“1”和“2”之间来回震荡，始终不能收敛。因此，Raghavan 等人[5] 又提出了标签的异步更新方法，在第 t 次迭代过程中，节点根据它的邻接点在当次迭代过程中已经完成标签更新的节点的新标签和还未进行更新的节点在 t-1 次迭代后得到的标签计算该节点的新标签。通过这种方式能够防止标签震荡现象的发生。

标签传播算法的设计方法简单，容易被人理解接受，算法的执行过程如算法2 所示。

算法：标签传播算法（LPA）

输入：复杂网络 $G=(V,E)$，最大迭代次数 $maxRun$

输出：社区划分结果

Step1: 初始化，为网络中的每个节点分配一个各不相同的标签，$c_i(0)=i$；令迭代次数 $t=0$。

Step2: 标签传播迭代过程

（a）如果迭代次数 $t>maxRun$，标签传播迭代过程结束，转 Step3；否则继续算法。

（b）随机排列网络中的节点，并将节点顺序存放在向量 X 中。

（c）对于每个节点 $v_i \in X$，更新 $c_i(t)=f(c_{i1}(t),c_{i2}(t),...,c_{im}(t))$ 表示在当次更新过程中节点 $v_{i1},...,v_{im}$ 的邻接点中标签已经更新的节点集，表示在当次更新过程中节点的邻接点中标签还未更新的节点集。这里的函数 $f(x)$ 将返回集合中出现频次最高的标签。如果返回不止一个标签，那么就在其中随机选择一个。

（d）如果所有节点的标签都不再改变，那么标签传播迭代过程停止，转 Step3；否则，令 $t=t+1$ 转到步骤（a）继续执行。

Step3: 社区划分，将拥有相同标签的节点划分到同一个社区中，不同标签的种类就表示网络中社区的个数。

LPA 算法每次迭代过程中更新节点的顺序随机确定，并且当多个标签在其邻接点中出现次数最多时，标签的更新也是随机的，因此每次执行标签传播算法都可能得到不同的社区划分结果。在众多的社区划分结果中，很难确定哪一个结果是最优的划分。所以解决标签传播算法的稳定性问题是非常有必要的，而且也是非常重要的。

3.2 CDABSLP 算法设计

3.2.1 k-核分解方法

在第二章第一小节社交网络的统计特性中，很多计算节点重要性的方法被提出，比如度、聚集系数和介数等。度和聚集系数仅能够衡量网络局部的信息；介数能够反映整个网络的全局信息，但是由于介数的计算需要计算网络中所有的最短路径，因此它的时间复杂度很高。Kitsak 等人 [?] 提出复杂网络中 k-核值高的节点对整个网络的信息传播是非常重要的，其传播能力强。

Algorithm 1 用归并排序求逆序数

Input: $Array$ 数组，n 数组大小
Output: 逆序数

```
1:  function MergerSort(Array, left, right)
2:      result ← 0
3:      if left < right then
4:          middle ← (left + right)/2
5:          result ← result+ MergerSort(Array, left, middle)
6:          result ← result+ MergerSort(Array, middle, right)
7:          result ← result+ Merger(Array, left, middle, right)
8:      end if
9:      return result
10: end function
11:
12: function Merger(Array, left, middle, right)
13:     i ← left
14:     j ← middle
15:     k ← 0
16:     result ← 0
17:     while i < middle and j < right do
18:         if Array[i] < Array[j] then
19:             B[k++] ← Array[i++]
20:         else
21:             B[k++] ← Array[j++]
22:             result ← result + (middle − i)
23:         end if
24:     end while
25:     while i < middle do
26:         B[k++] ← Array[i++]
27:     end while
28:     while j < right do
29:         B[k++] ← Array[j++]
30:     end while
31:     for i = 0 → k − 1 do
32:         Array[left + i] ← B[i]
33:     end for
34:     return result
35: end function
```

Algorithm 2 Framework of ensemble learning for our system.

Input: The set of positive samples for current batch, P_n; The set of unlabelled samples for current batch, U_n; Ensemble of classifiers on former batches, E_{n-1};
Output: Ensemble of classifiers on the current batch, E_n;
1: Extracting the set of reliable negative and/or positive samples T_n from U_n with help of P_n;
2: Training ensemble of classifiers E on $T_n \cup P_n$, with help of data in former batches;
3: $E_n = E_{n-1} cup E$;
4: Classifying samples in $U_n - T_n$ by E_n;
5: Deleting some weak classifiers in E_n so as to keep the capacity of E_n;
6: **return** E_n;

k-核分解方法是将复杂网络分解成若干子结构，子结构中的每个节点在该子结构中的度最小为 k。为 k-核中的每个节点 i 分配一个 k-核值，用 Ks(i) 表示，该值表明节点 i 属于 k-核，但是不属于 (k+1)-核。k-核值的大小反应了节点在网络中的中心性地位，k-核分解方法经常被用于识别网络中的中心和边缘节点。

k-核分解方法的具体步骤如下：首先不断的将网络中度为 1 的节点及与这些节点相连的边移除，直到剩余网络中不再有度为 1 的节点为止，并将移除的节点划分为 1-核子集，并设置这些节点的 k-核值为 1；使用同样的方法，递归的移除剩余网络中度为 2 或小于 2 的节点及连接这些节点的边，直到剩余网络中不再有度为 2 或小于 2 的节点为止，创建 2-核子集；分解过程继续执行，直到网络中所有的节点都被划分到相应的 k-核子集中。k-核值大（或小）的节点子集位于网络的中心（或边缘）位置。通过 k-核分解方法能够得到网络的层次结构，该结构类似于一个洋葱，反应了网络中节点完整的层次结构。k-核分解方法能够在线性时间内执行完成，时间复杂度为 $O(|E|)$，其中 |E| 是网络中边的数量。

3.2.2 异步标签传播策略

异步标签传播策略能够避免标签震荡现象，并且相对同步标签更新策略需要更少的迭代次数，因此这里采用异步标签更新方法。然而由于节点并不是同时更新的，因此节点更新的顺序对社区发现结果的稳定性及社区质量有很大的影响；除此之外，标签传播过程中的标签选择策略也存在不稳定因素，当返回多个标签同时被最大个数邻接点拥有时，LPA 算法随机的选择其中的一个标签作为该节点的新标签，这也造成了 LPA 算法的不稳定性。而下一章节要提到的 COPRA 中也同样存在这些不稳定因素。

在简单网络上分析传统标签传播算法的社区发现过程，如图3.2所示。在该网络中

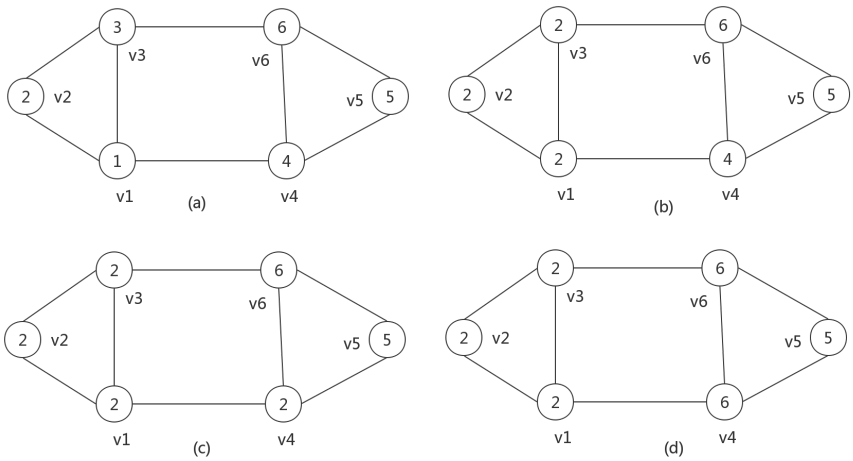

图 3.2　标签传播过程示意图

有两个社区，分别是 v_1, v_2, v_3 和 v_4, v_5, v_6。节点上的数字表示该节点的社区标签，初始的时候各个节点的社区标签各不相同，如图3.2(a) 所示。假设经过几次标签传播之后，节点拥有相同的标签“2”，而节点 v_4、v_5 和 v_6 的标签仍然各不相同，如图3.2(b) 所示。如果首先更新节点 v_4 的标签，由于它的所有邻接点的标签都各不相同，随机选择标签“2”作为节点 v_4 的新标签，更新结果如图3.2(c) 所示；然后更新节点 v_6 的标签为标签“2”，此时节点 v_5 的两个邻接点的标签都为“2”，它也更新为标签“2”。这样更新后所有的节点都划分到了同一个社区中，这样的社区划分结果没有意义。相反，在更新节点 v_4 的标签时，如果随机选择的更新标签是标签“6”，如图 3.2(d) 所示；紧接着更新节点 v_5 的标签，根据标签计算公式得到节点 v_5 的新标签为标签“6”；此时节点 v_6 的两个邻接点的标签都为“6”，它的标签保持不变。通过这样的标签更新顺序和标签选择方式，能够得到正确的社区划分结果。

通过上面的分析，传统的异步标签传播算法对节点更新的顺序及标签选择方法非常敏感，标签传播过程中的随机性不仅造成算法收敛速度的不同，甚至会影响最终的社区划分结果。因此，本章提出一种改进的标签传播算法，克服传统标签传播算法的不足。

3.2.3　核心思想

在 CDABSLP 算法中，仍采用异步更新策略来避免图3.1所示的标签震荡现象的出现。但是，不确定的节点更新顺序导致算法的稳定性差，新算法中需要克服此问题。在算法每次的更新过程中，先更新的节点的标签在整个标签传播过程中发挥的作用比

后更新的节点的标签的作用要大，这是因为后更新节点的新标签对已经更新的节点的标签没有影响，最后一个更新的节点的新标签对其他所有节点的标签选择都不会产生影响。因此，算法应该根据节点的重要性对节点进行排序，重要的节点应该优先更新。

k-核值大的节点表示它位于网络的核心位置，然而，在复杂网络中，有大量节点拥有相同的 k-核值，因此，仅根据该指标对节点进行排序的效果并不好。通常，在复杂网络中，一个节点如果和很多核心节点相连，那么该节点在网络中的地位也是很重要的。受该思想的启发，本章提出一种同时考虑节点本身 k-核值和其邻接点 k-核值共同影响的节点中心性衡量指标。节点 i 的影响值的计算如公式3.2所示。

$$NI(i) = Ks(i) + \alpha \times \sum_{j \in \Gamma_i} \frac{Ks(j)}{d_j} \tag{3.2}$$

其中，α 是调节参数，取值范围从 0 到 1，用来调节邻接点对节点影响值的作用大小。将公式3.2计算得到的节点影响值作为衡量节点重要性的指标，按节点影响值降序对节点进行排序作为节点更新的顺序。固定节点更新顺序能够使算法更加稳定。

造成标签传播算法不稳定的另一个因素是标签选择的机制，当更新一个节点的标签时，如果返回多个标签同时被最大个数的邻接点拥有时，传统的标签传播算法会随机的从中选择一个标签赋给该节点，因此，算法迭代过程很难得到一个稳定的收敛状态。为了提高算法的稳定性，当返回多个标签时，将节点影响值引入到标签更新公式中，选择标签影响强度最大的标签赋给该节点。

标签 l 对节点 i 的影响强度计算如公式3.3所示。

$$NI(i,l) = \sum_{j \in \Gamma_i} \frac{NI(j)}{d_j} \tag{3.3}$$

其中，Γ_i 表示节点 i 的邻接点中标签为 l 的节点集合。改进的节点标签更新公式如公式3.4所示。

$$c_i = \arg \max_{l \in lmax} LI(i,l) \tag{3.4}$$

其中，$lmax$ 表示同时被最大个数邻接点拥有的标签集合。当传统标签传播算法的标签更新公式返回多个标签时，根据公式3.3计算这些标签对该节点的影响强度，选择影响强度最大的标签赋给该节点。当标签影响强度最大的标签仍有多个时，节点保

留原有标签。

3.2.4 执行步骤

CDABSLP 算法的主要步骤包括初始化、迭代标签传播和社区划分，图3.3为 CDABSLP 算法的流程图。

通过一个简单的例子来展示 CDABSLP 算法的执行过程，如图3.4所示，算法中参数 $\alpha = 1$。图中每一个圆圈代表一个节点，节点间的连线代表节点间的边，节点外的实数表示节点的影响值 NI。按节点影响值降序排列图中的节点 $v_1 - v_2 - v_4 - v_6 - v_5$（当节点影响值相同时，按节点的先后顺序排列），以该顺序作为节点更新的顺序，标签更新过程如图3.4所示。

按照节点更新顺序，第一个更新节点 v_1 的标签。首先为节点 v_1 计算一系列的三元组 $(l, |\Gamma_1^l|, LI(v_1, l))$，其中 l 表示其邻接点中包含的标签，$|\Gamma_1^l|$ 表示标签为 l 的邻接点的个数，最后一项 $LI(v_1, l)$ 表示标签对该节点的影响强度，该项是一个可选项，当不能通过传统的标签选择策略得到一个确定的标签时，通过公式3.3 计算得到。如图3.4(a) 所示，节点 v_1 有三个邻接点 v_2 、v_3 和 v_4，并且它们的标签各不相同，计算得到节点 v_1 对应的三元组集合为 $(2, 1, 1.833), (3, 1, 1.667), (4, 1, 1.667)$。因此选择标签 2 作为节点 v_1 的新标签。

接着更新节点 v3 的标签。更新完节点 v_1 的标签之后，如图3.4(a) 右图所示，节点 v_3 共有三个邻接点 v_1、v_2 和 v_6，其中 v_1 和 v_2 的标签相同，均为标签 2，只有节点 v_6 的标签不同。因此选择标签 2 作为节点 v_3 的新标签，如图3.4(b) 所示。接下来节点 v_4 和 v_6 标签的更新分别与节点 v_1 和 v_3 的情况相同，更新结果如图3.4(c) 所示。

最后只有节点 v_2 和 v_5 没有更新，此时如图3.4(c) 所示，节点 v_2 和它的邻接点的标签都为标签 2，而节点 $v5$ 与它所有的邻接点的标签都为标签 5，因此节点 v_2 和 v_5 的标签不需要改变。

通过执行 CDABSLP 算法，在该网络上仅需执行一次标签更新过程就得到了最终稳定的社区划分结构，得到两个与真实情况一致的社区。由于算法的执行过程中没有了随机因素的存在，因此算法的输出结果是确定的且优质的。

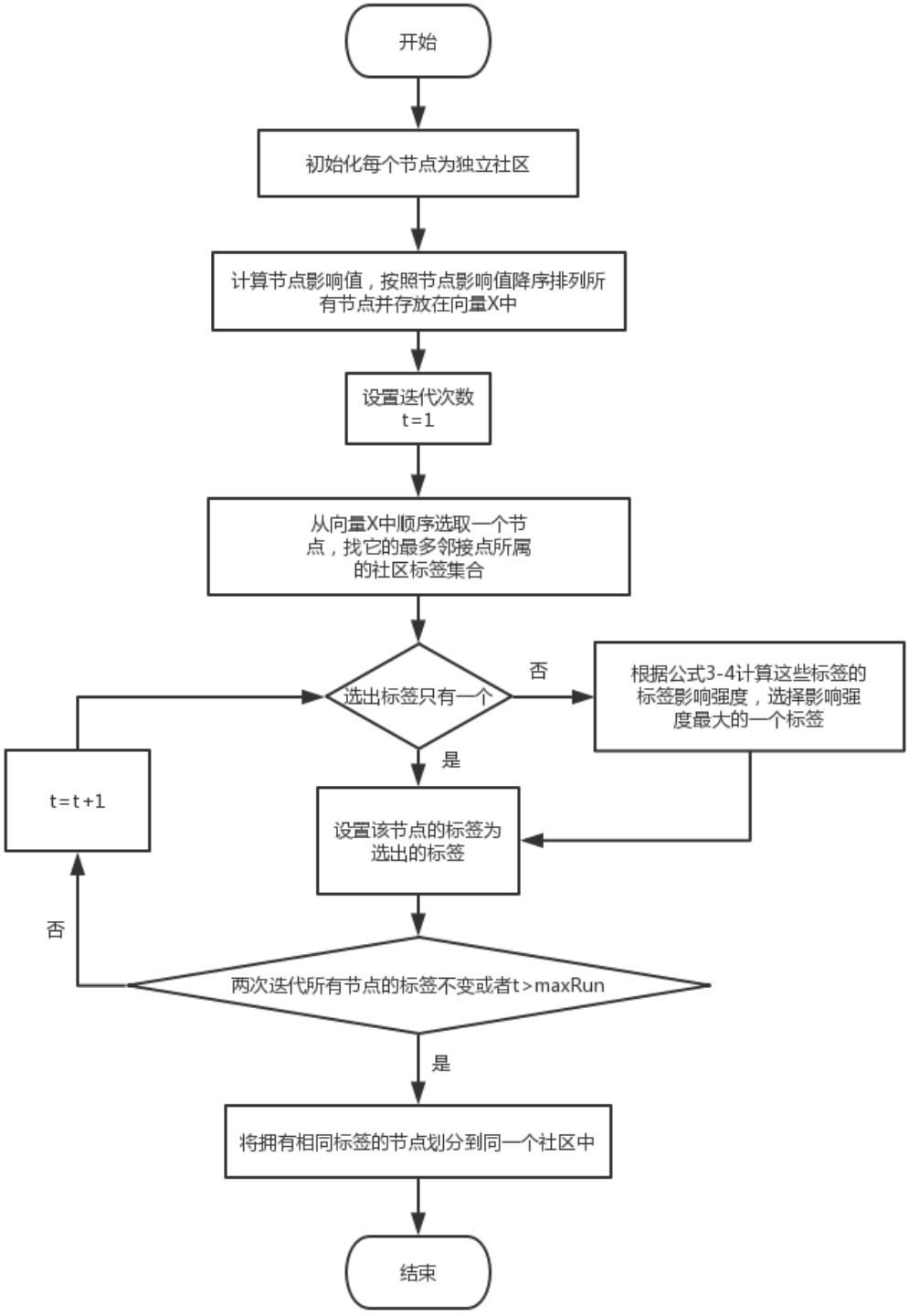

图 3.3　CDABSLP 算法流程图

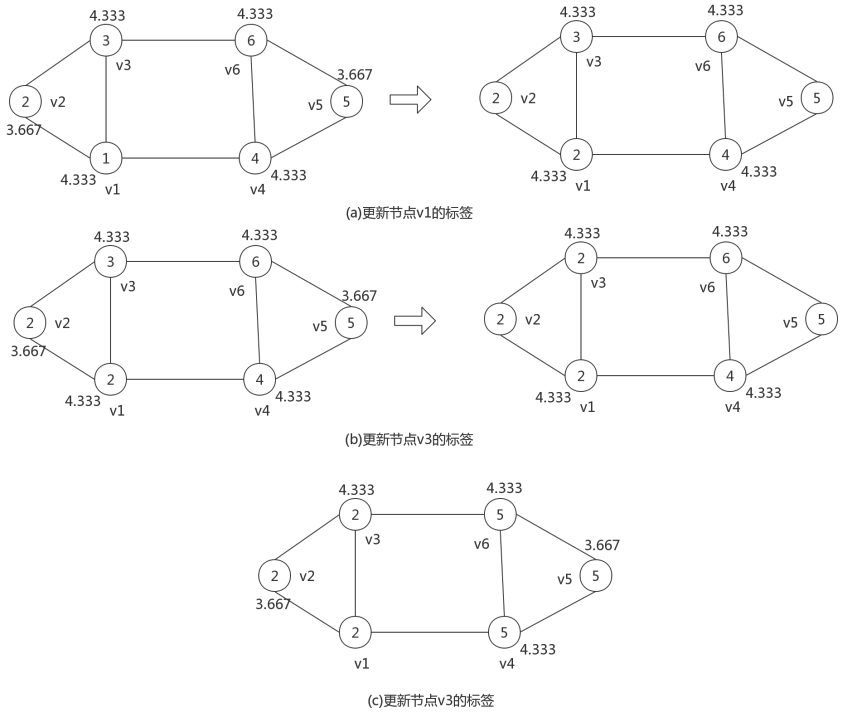

图 3.4　CDABSLP 算法标签传播过程示意图

3.2.5　时间复杂度分析

算法的时间复杂度分析如下，$|V|$ 表示网络中节点的个数，$|E|$ 表示网络中边的数目。

（1）为每个节点初始化标签所用时间复杂度为 $O(|V|)$；

（2）计算网络中所有节点的影响值的时间复杂度为 $O(|E|)$；

（3）按节点影响值降序排列网络中所有节点所用时间复杂度为 $O(|V|log(|V|))$；

（4）每次标签传播过程分为两部分：传统的标签计算过程：$O(|E|)$；当传统标签计算过程返回多个标签时，利用公式3.3重新计算节点标签的过程：$O(|E|)$；

（5）将相同标签的节点划分到一个社区的时间复杂度为 $O(|V|)$。

标签传播过程是不断迭代执行的，因此整个算法的时间复杂度为 $2O(|V|)+(2t+1)O(|E|)+O(|V|log(|V|))$，$t$ 表示迭代次数，一般通过比较少的迭代次数就能得到最后的结果。

表 3.1　计算机硬件配置

处理器	2.2GHz 双核 Intel Core i7
内存容量	8 GB 1600 MHz DDR3
硬盘容量	128GB 固态硬盘

表 3.2　计算机软件配置

操作系统	macOS Sierra 10.12.6
Anaconda 版本	conda 4.3.30
Networkx 版本	2.1
Python 版本	2.7.14
Matplotlib 版本	2.0.2
Numpy 版本	1.13.1

3.3　CDABSLP 算法验证实验

本节为本章提出的基于稳定标签传播的非重叠社区发现算法 CDABSLP 算法进行实验验证。首先介绍实验的软硬件环境和采用的数据集，然后对算法的评价指标进行简单阐述，最后是相关对比实验的结果展示与分析。

3.3.1　实验环境

本文实现的 CDABSLP 算法所使用的机器配置如表3.1所示。CDABSLP 算法使用 Python 语言编程实现，均基于 Python 的复杂网络相关软件包 Networkx，使用 Anaconda 来对软件包进行管理和部署，具体配置如表3.2所示。

3.3.2　数据集

选用 5 个不同的真实数据集和 LFR 基准网络人工生成数据集进行实验验证本章所提算法的有效性。

（1）真实数据集

在 5 个常用的真实网络数据集上进行实验验证本章算法的有效性，这 5 个真实网络数据集包括 Karate、Dolphins 和 Football 等，各个数据集的详细信息如表3.3所示；

R1:Karate 是 Zachary 空手道俱乐部成员关系网络，网络中的所有节点对应各个成员，边表示两个端点对应的成员是好朋友。网络包含 34 个节点，78 条边和两个社区。

R2:Dolphins 是 Lusseau 等人对栖息在新西兰 Doubtful Sound 峡湾的一个宽吻海豚

表 3.3　真实网络数据集

数据集名称	节点数	边数	社区数
Karate	34	78	2
Dolphins	62	159	2
Polbooks	105	441	3
Football	115	616	12
Email	1133	5451	

群体进行长达 7 年的观察所构造出的海豚关系网，该群体包含 2 个家族共 62 只宽吻海豚。由这个群里的所有成员及它们间的接触关系构成一个包含 62 个节点，159 条边和两个社区的网络。

R3:Polbooks 是从 Amazon 的图书销售记录抽象得到的网络数据集，分析了 105 本与美国政治相关的书和它们的 441 条共同销售关系，依据亚马逊上对图书的观点和评价情况，将这些书分为“自由派”、“中间派”和“保守派”三个类。因此，此数据集包含 105 个节点，441 条边和三个社区。

R4:Football 是分析美国高校橄榄球比赛对阵表得到的数据集。共有 115 所高校派出代表队参赛，共进行了 616 场比赛，按各代表队地区的不同将这个包含 115 个节点 616 条边的网络分为 12 个社区。

R5:Email 是由 Guimer 等人收集公布的，包含位于西班牙加泰罗尼亚自治区的罗维拉-威尔吉利大学（简称 URV）的教师和研究生之间的邮件往来关系。两个用户或者说两个邮箱地址如果互相发送过邮件，就构成一条边。网络包含 1133 个节点和 5451 条边。

（2）LFR 人工基准网络

LFR 基准网络是目前在社区发现领域使用最多的人工数据集之一。通过调整网络生成参数可以产生用户需要的不同的人工数据集，LFR 基准网络的主要生成参数及其含义如表3.4所示。

在 LFR 模型众多的生成参数中，混合参数 $mu \in [0,1]$ 是非常重要的一个参数，mu 越小，说明连接社区之间的边越少，社区之间越“分离”，社区划分的难度随着 mu 的增长而增大。on 和 om 两个参数用于生成具有重叠社区的数据集，生成具有非重叠社区结构的数据集时，只需将 on 设置为 0，om 设置为 1 即可。生成六组具有非重叠社区结构的 LFR 基准网络数据集，所有的网络共享的相同参数是 maxk = 50、on = 0 和 om = 1。每组包含九个 mu 值不同的数据集，分别为 0.1 到 0.9，每组中的九个数据

表 3.4　LFR 基准网络生成参数及其含义

参数	含义
N	节点数
avgk	节点平均度
maxk	节点最大度
mu	网络拓扑结构混合参数
minc	最小社区规模
maxc	最大社区规模
on	重叠节点个数
om	重叠节点可属于的社区个数

表 3.5　六组 LFR 基准网络生成参数

编号	N	avgk	maxk	minc	maxc	mu
S1	100000	100	5000	100	5000	$0.1 \sim 0.9$
S2	100000	100	5000	200	1000	$0.1 \sim 0.9$
S3	500000	100	5000	100	5000	$0.1 \sim 0.9$
S4	500000	100	5000	200	1000	$0.1 \sim 0.9$
S5	100000	200	5000	100	5000	$0.1 \sim 0.9$
S6	100000	200	5000	200	1000	$0.1 \sim 0.9$

集共享参数 N、avgk、minc 和 maxc。其他参数都取默认值。表3.5展示了这六组网络详细的生成参数情况。

3.3.3　评价指标

迄今为止，出现了各种各样的社区发现算法，如何评价不同的的发现算法的好坏是一个非常重要的问题。为此，学者们提出了多种社区结构评价指标用来评价网络社区划分质量，其中比较有代表性的有模块度、NMI 等。下面介绍这些指标。

（1）模块度

模块度是目前学者们最常用和经典的网络社区结构评价指标，它最初是被 Newman 等人于 2004 年提出来的[3]。其通过比较现有网络和基准网络在相同社区划分下的连接密度差来衡量网络社区的优劣，其中基准网络是由原网络具有相同度序列的随机网络。模块度计算方式详见公式3.5。

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{3.5}$$

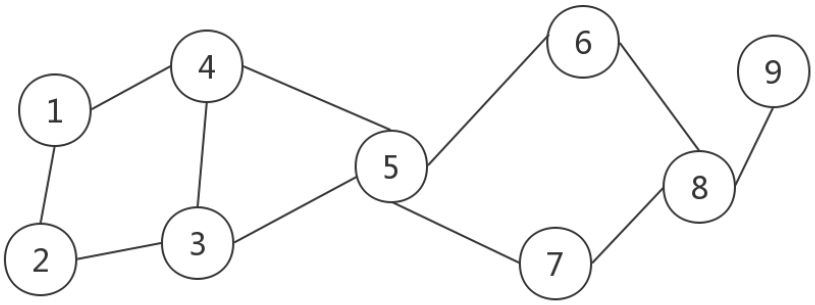

图 3.5　NMI 网络示例图

其中，A 表示网络中的邻接矩阵，m 表示网络中边的总数，k_i 和 k_j 表示节点 i 和 j 的度数，c_i 和 c_j 表示节点 i 和 j 所属的社区。如果 $i=j, \delta(c_i,c_j)=1$，反之 $\delta(c_i,c_j)=0$

（2）NMI

随着在线社交网络的发展，人们发现在线社交网络的很多数据中存在着暗示各个节点的社区属性信息。例如，在人人网的学校信息便揭示了网络节点中属于同一学校的社区结构，Facebook 中的兴趣信息同样表征了具有相同兴趣的虚拟用户群体。这些数据在为社区发现问题提供了丰富的信息的同时，也在一定程度上为虚拟社区结构优劣的评判提供了标准答案。针对这种预先拥有一定虚拟社区结构信息的情况下，Leon Danon 等人 [?] 提出了 Normalized Mutual Information（NMI）利用信息化熵来衡量算法划分的社区结构和预先已知的社区结构之间的差异。NMI 是基于混合矩阵（Confusion Matrix）N 来计算的数字指标。NMI 计算方式详见公式3.6。

$$NMI=\frac{-2\sum_{i,j}N_{ij}ln\frac{N_{ij}}{N_iN_j}}{\sum_i N_i ln\frac{N_i}{n}+\sum_j N_j ln\frac{N_j}{n}} \tag{3.6}$$

使用该数字指标，可以衡量划分出来的社区结构与已知的网络社区结构的差异程度值，该值越大，则表明获得的社区结构划分越好，当该值达到最大化值 1 时，说明算法发现的社区结构与已知社区结构完全已知，效果最好。

下面以图3.5为例来说明计算 NMI 的过程。假设已知的最佳社区结构划分为集合 1，2，3，4 和 5，6，7，8，相应的社区划分向量表示为 a = (1,1,1,1,2,3,3,3,3)，再假设某算法获得的社区划分结构可以用向量表示为 b = (3,3,3,3,2,1,1,1,1) 来表示。根据已知的社区划分向量，可以构造混合矩阵3.7。

$$N = \begin{bmatrix} 0 & 0 & 4 \\ 0 & 1 & 0 \\ 4 & 0 & 0 \end{bmatrix} \tag{3.7}$$

根据上式计算可知，该划分的 NMI 值为 1。

（3）PWF

成对 F-measure（Pairwise F-measure，PWF）是成对准确率（Pairwise Precision，PWP）和成对召回率（Pairwise Recall，PWR）的调和，其计算如公式3.8所示。

$$PWF = \frac{2 \cdot PWP \cdot PWR}{PWP + PWR} \tag{3.8}$$

成对准确率（PWP）和成对召回率（PWR）的计算公式分别为公式3.9和公式3.10。

$$PWP = \frac{|S \cap T|}{|S|} \tag{3.9}$$

$$PWP = \frac{|S \cap T|}{|T|} \tag{3.10}$$

T 表示在真实的社区划分结果中，在同一个社区内的节点对集合；S 表示在测试算法得到的社区划分结果中在同一个社区内的节点对集合；$|S \cap T|$ 表示在真实的社区划分结果和测试划分结果中都在同一个社区内的节点对集合。PWF 的取值范围是 $0 \sim 1$，PWF 越大，说明社区划分的准确率越高。

3.3.4 实验结果及分析

(1)LFR 基准网络上的实验

图3.6 ～ 3.11分别是四种算法在六组非重叠 LFR 基准网络数据集（S1∼S6）上实验结果的 NMI 和 PWF 指标的对比图。横轴代表混合参数 mu，取值从 0.1 到 0.9；左侧六幅图的纵轴代表社区划分结果的 NMI 值，右侧六幅图的纵轴表示实验结果的 PWF 值。

图3.6 ～ 3.11中的 12 幅图可以看出，随着 mu 值的增大，网络的结构越来越复杂，社区结构越来越不明显，四种算法得到的社区划分结果都随之变差，尤其是当 mu 大

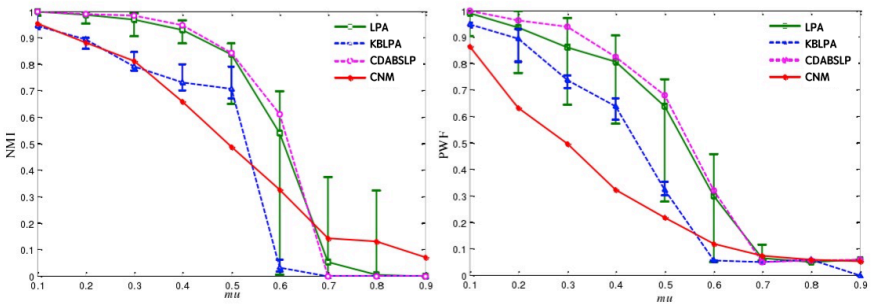

图 3.6　在 S1 网络上的实验结果的 NMI 和 PWF 比较

表 3.6　真实网络的模块度实验结果

编号	LPA	KBLPA	CDABLPA	CNM
R1	0.296 ± 0.29	0.073 ± 0.23	0.423	0.345
R2	0.465 ± 0.19	0.489 ± 0.12	0.521	0.306
R3	0.489 ± 0.15	0.449 ± 0.09	0.497	0.265
R4	0.582 ± 0.14	0.573 ± 0.09	0.582	0.537
R5	0.380 ± 0.27	0.183 ± 0.34	0.427	0.415

于 0.5 时，NMI 和 PWF 指标下降的更快。但是，整体来看，CDABSLP 算法的效果优于其他三种算法。虽然 CDABSLP 算法并不是在所有情况下都能得到最优的结果，但是它得到的结果是稳定的并且比较好的。从对比图中还能看出 LPA 算法得到结果的 NMI 和 PWF 的波动都很大；KBLPA 算法得到的结果是比较稳定的，但是该算法检测得到的社区结构比 LPA 和 CDABSLP 算法都要差一些。在所有这些 LFR 网络上，CNM 算法并不能检测到最优的社区结构，而且它得到的社区的数目通常都比真实情况少。

（2）真实网络上的实验

在数据集中介绍的 5 个真实网络数据集经常出现在社区发现的文献中，使用模块度 Q 和标准化互信息 NMI 作为前四个数据集上实验结果的评价指标，而另外一个数据集上仅使用模块度 Q 作为评价指标。表3.6和表3.7给出了四种对比算法在 5 个真实网络数据集上的实验结果，由于 R5 网络的真实社区结构未知，所以在表 3-8 中仅给出了前四个网络上实验结果的 NMI 值。每个数据集上得到的最优的 Q 和 NMI 值用粗体表示，LPA 算法和 CDABSLP 算法得到结果的 Q 和 NMI 以平均值 ± 最大偏差的形式表示。

从表3.6和表3.7可以看出，CDABSLP 算法得到社区结构的模块度 Q 比其他三种算法都高。同时，CDABSLP 算法在前四个网络上得到的社区结构的 NMI 值也是最优

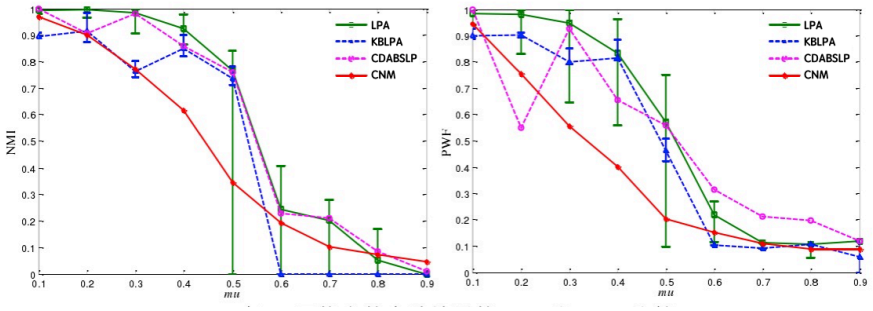

图 3.7　在 S2 网络上的实验结果的 NMI 和 PWF 比较

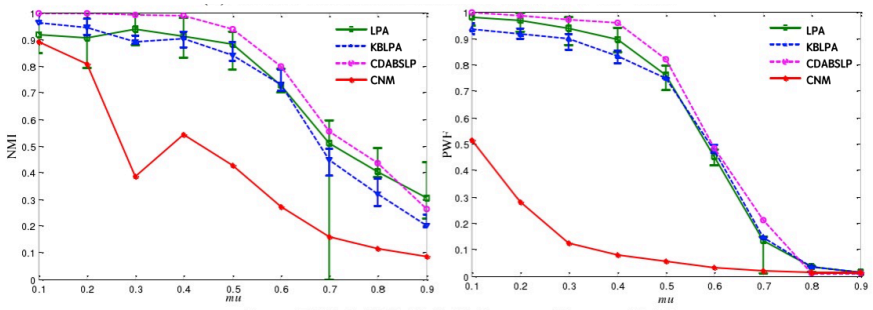

图 3.8　在 S3 网络上的实验结果的 NMI 和 PWF 比较

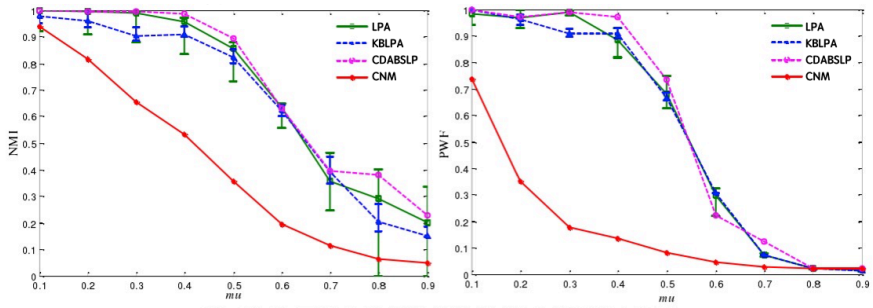

图 3.9　在 S4 网络上的实验结果的 NMI 和 PWF 比较

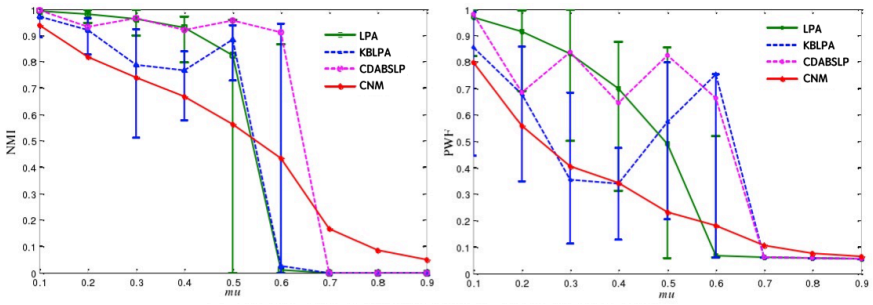

图 3.10　在 S5 网络上的实验结果的 NMI 和 PWF 比较

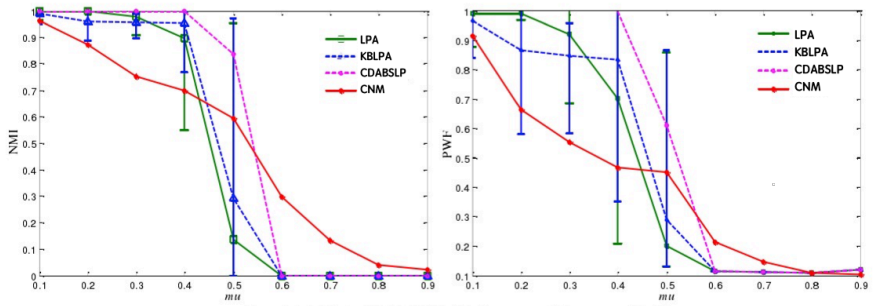

图 3.11　在 S6 网络上的实验结果的 NMI 和 PWF 比较

表 3.7　真实网络的模 NMI 实验结果

编号	LPA	KBLPA	CDABLPA	CNM
R1	0.583 ± 0.58	0.173 ± 0.53	1	0.435
R2	0.526 ± 0.19	0.471 ± 0.22	0.622	0.426
R3	0.574 ± 0.07	0.524 ± 0.09	0.656	0.255
R4	0.863 ± 0.24	0.849 ± 0.14	0.878	0.636

的。KBLPA 算法的稳定性优于 LPA 算法，但是 KBLPA 算法在几乎所有网络上得到社区结构的 Q 和 NMI 都没有 LPA 算法好。实验结果表明，CDABSLP 算法能够得到比其他三种算法更好更稳定的社区检测结果。

（3）可视化对比

待加入。。。

3.3.5　实验总结

CDABSLP 算法从两个方面改善 LPA 算法不稳定的问题：首先，计算网络中每个节点的节点影响值并按节点影响值降序排列作为节点标签更新的顺序，取代传统 LPA 算法中节点更新顺序随机确定的方法；其次，在每次标签更新迭代过程中，当传统的标签计算方法返回多个标签时，提出一种新的标签计算公式，计算返回标签的影响强度，在返回的多个标签中重新选择一个影响强度最大的标签作为该节点的新标签，以此替代传统 LPA 算法中随机选择一个标签的方法。CDABSLP 算法既保持了传统 LPA 算法的优点，还解决了 LPA 算法不稳定的问题，该算法能够得到稳定的社区结构。大量实验结果表明 CDABSLP 算法的性能优于目前一些代表性的社区发现算法。

第 4 章　基于稳定标签传播的重叠社区发现算法

在验证了上一章所提稳定策略在 LPA 算法上的有效性的基础上，将此稳定策略运用到 COPRA 算法中，以验证其在重叠社区发现算法中的有效性。COPRA 算法 [?] 和 SLPA 算法 [?] 通过允许每个节点拥有多个标签的方法，将 LPA 算法扩展应用于重叠社区的发现，它们既继承了 LPA 算法的优点，也保留了 LPA 算法不稳定和鲁棒性差等缺点。COPRA 算法是最早的基于标签传播的重叠社区发现算法。

本章提出一种基于稳定标签传播的重叠社区发现算法（Overlapping Community Detection Algorithm Based on Stable Label Propagation）, 下文简称 OCDABSLP。OCDABSLP 算法在迭代执行标签更新过程中，当节点属于所有社区的隶属度都小于阈值且最大值有多个时，选择隶属度最大的多个标签中标签影响强度最大的标签。满足终止条件后，算法根据节点的标签将其划分到相应社区中，拥有多个标签的节点被划分到相应的多个社区中，成为重叠节点，得到最终的重叠社区划分结果。在不同复杂网络数据集上的大量实验表明本章算法能够得到比现有的大部分算法更好的社区划分结果。

本章接下来的内容组织结构上将先简单介绍下多标签传播算法（COPRA），然后详细介绍 OCDABSLP 算法的设计思路、核心思想和关键步骤等，最后在真实网络以及人工基准网络上的实验，并与其他基准算法进行对比实验，以此来分析算法的效果。

4.1　多标签传播算法 (COPRA)

Gregory 等人 [?] 提出的 COPRA 算法是第一个利用标签传播思想进行重叠社区发现的算法。算法中每个节点可以以不同隶属度拥有多个标签，每个节点包含一组标签-隶属度对 (l, b)，l 表示节点所属社区的编号，b 表示节点属于该社区的隶属程度，$b_t(l, i)$ 表示在第 t 次标签传播结束时节点 i 属于社区 l 的隶属程度。COPRA 通过迭代地更新各个节点的标签及隶属度来获取社区结构。

与 LPA 算法相同，初始时，COPAR 算法为每个节点分配一个各不相同的标签，并将其隶属度设置为 1，即 $b_0(i, i) = 1$。然后采用同步更新策略进行标签更新迭代，在每次更新过程中，用邻接点中出现的所有相同标签的平均隶属度更新该节点的标签-隶

属度对列表。每一轮更新后，删除隶属度小于 $\frac{1}{v}$ 的标签（v 是算法的参数），当一个节点的所有标签对应的隶属度都小于 $\frac{1}{v}$ 时，就只保留一个隶属度最大的标签，若此时有多个标签的隶属度同时取最大值，就随机保留隶属度最大的标签中的一个，然后对所有剩余标签的隶属度进行归一化。更新结束后，算法根据节点的标签将其划分到相应的社区中。一个节点最后拥有的标签数即为它被划分到的社区的个数。

函数 $b_t(l,i)$ 用于计算在第 t 次迭代中，节点 i 属于社区 l 的隶属程度，计算如公式4.1所示。

$$b_t(l,i) = \frac{\sum_{j\in\Gamma_i} b_{t-1}(l,j)}{d_i} \tag{4.1}$$

COPRA 算法的执行过程如算法所示。

算法：多标签传播算法（COPRA）

输入：复杂网络 $G=(V,E)$，最大迭代次数 $maxRun$

输出：社区划分结果

Step1: 初始化，为网络中的每个节点分配一个各不相同的标签，标签-隶属度对集合为 (i,l)；令迭代次数 $t=0$。

Step2: 标签传播迭代过程

（a）如果迭代次数 $t > maxRun$，标签传播迭代过程结束，转 Step3；否则继续算法。

（b）对于每个节点 $v_i \in V$，根据公式4.1计算该节点属于其邻接点集合中出现的所有标签的隶属度，更新标签-隶属度对列表。根据参数 v 删除不满足条件的标签，并对剩余标签进行归一化。

（c）如果连续两次迭代结束后，标签集合的大小不变，那么标签传播迭代过程停止，转 Step3；否则，令 $t=t+1$ 转到步骤（a）继续执行。

Step3: 社区划分，将拥有相同标签的节点划分到同一个社区中，不同标签的种类就表示网络中社区的个数。

COPRA 算法继承了 LPA 算法的优点，也保留了 LPA 算法稳定性和鲁棒性差等缺点。

4.2 OCDABSLP 算法设计

4.2.1 核心思想

原始 COPRA 算法中的唯一一个不稳定因素是当节点属于所有社区的隶属度都小于阈值且最大值有多个时，会随机选择一个标签。因此，在此处对 COPRA 算法进行改进。当出现上述情形时，选择隶属度最大的多个标签中标签影响强度最大的标签。当标签影响强度最大的标签仍有多个时，保留所有标签影响强度最大的标签。标签影响强度的计算如公式4.2所示。

$$NI(i,l) = \sum_{j \in \Gamma_i} b_{t-1}(l,j) \frac{NI(j)}{d_j} \tag{4.2}$$

4.2.2 执行步骤

OCDABSLP 算法的主要步骤包括初始化、迭代标签传播和社区划分。图4.1 为 OCDABSLP 的算法流程图。

4.2.3 时间复杂度分析

OCDABSLP 算法的时间复杂度分析如下：

（1）为每个节点初始化标签所用时间复杂度为 $O(|V|)$；

（2）每次标签传播过程分为两部分：传统的标签传播过程：$O(v|E|log(v|E|/|V|))$；当节点属于所有社区的隶属度都小于阈值且最大值有多个时，利用公式4.2计算标签影响值的过程：$O(v|E|log(v|E|/|V|))$；

（3）将相同标签的节点划分到一个社区的时间复杂度为 $O(|V|)$。

标签传播过程是不断迭代执行的，因此整个算法的时间复杂度为 $2O(|V|)+2tO(v|E|log(v|E|/|V|)$

4.3 OCDABSLP 算法验证实验

本节为本章提出的基于稳定标签传播的重叠社区发现算法 OCDABSLP 算法进行实验验证。首先介绍实验的软硬件环境和采用的数据集，然后对算法的评价指标进行简单阐述，最后是相关对比实验的结果展示与分析。

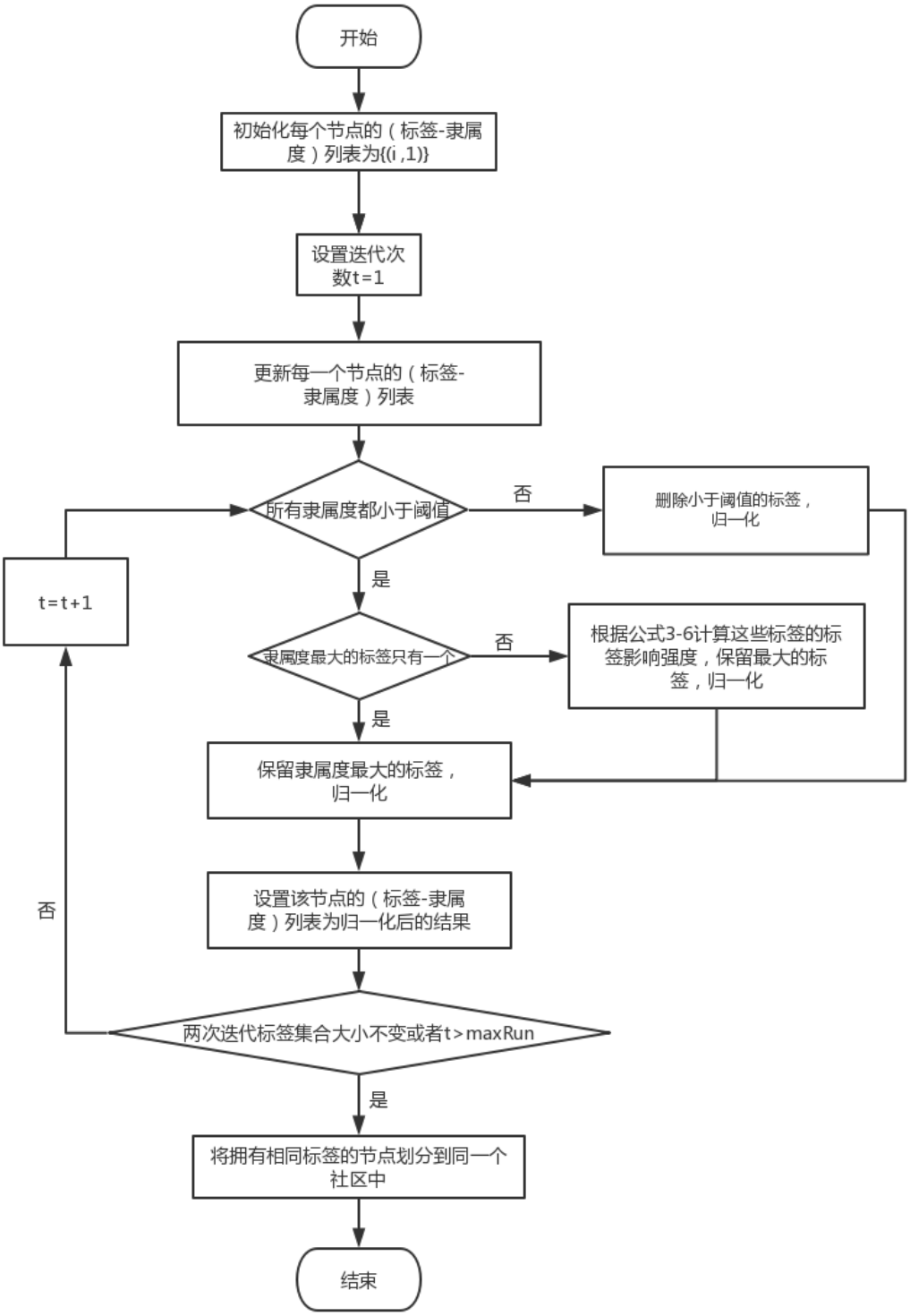

图 4.1　OCDABSLP 算法流程图

表 4.1　四组重叠 LFR 基准网络生成参数

编号	N	avgk	maxk	minc	maxc	mu	on	om
S7	100000	150	5000	100	5000	0.1	1000	2 ∼ 8
S8	100000	150	5000	100	5000	0.3	1000	2 ∼ 8
S9	500000	150	5000	200	10000	0.1	1000	2 ∼ 8
S10	500000	150	5000	200	10000	0.3	1000	2 ∼ 8

4.3.1　实验环境

本文实现的 OCDABSLP 算法所使用的软硬件环境与上一章节的 CDABSLP 算法一致，机器配置如表3.1所示。OCDABSLP 算法使用 Python 语言编程实现，均基于 Python 的复杂网络相关软件包 Networkx，使用 Anaconda 来对软件包进行管理和部署，具体配置如表3.2所示。

4.3.2　数据集

选用 4 组不同的 LFR 基准网络人工生成数据集进行实验验证本章所提算法的有效性。

（2）LFR 人工基准网络

LFR 基准网络是目前在社区发现领域使用最多的人工数据集之一。通过调整网络生成参数可以产生用户需要的不同的人工数据集，LFR 基准网络的主要生成参数及其含义在上一章节中已经提及，如表3.4所示。

本节实验将生成四组具有重叠社区结构的 LFR 基准网络数据集，详细的生成参数如表4.1示。

4.3.3　评价指标

本章采用重叠 NMI（ENMI）[?] 和重叠模块度 EQ [?] 作为重叠社区发现结果的评价指标。下面介绍这些指标。

（1）EQ

在上一章节已经提到了模块度的概念，而重叠社区模块度（EQ）[?] 可以在其公式3.5的基础上修改为公式4.3。

$$Q = \frac{1}{2m}\sum_{k=1}^{c}\sum_{i,j\in C_k}\frac{1}{O_i O_j}\left[A_{ij} - \frac{k_i k_j}{2m}\right] \tag{4.3}$$

在公式4.3中，i 和 j 表示网络中的两个节点，m 表示网络中边的数量，k_i 和 k_j 表示节点 i 和 j 的度数，O_i 和 O_j 表示节点 i 和 j 所属的社区的个数，A 表示网络的邻居矩阵，C_k 表示网络的第 k 个社区。该公式的数学意义为：网络中同一社区内部的边的比例与在同样社区结构下的基准网络内部边的比例的期望值之差。模块度越高，则网络中社区划分结果越好。

（2）ENMI

在上一章节已经提到了 NMI 的概念，而 ENMI[?] 是在其重叠社区上的扩展，具体计算方式详见公式4.4。

$$NMI(X,Y) = 1 - \frac{1}{2}[H(X|Y)_{norm} + H(Y|X)_{norm}] \tag{4.4}$$

其中 $H(X,Y)$ 函数表示联合熵，X 和 Y 分别是一个社区，$H(X \mid Y)$ 函数表示条件熵。

4.3.4 实验结果及分析

（1）人工基准网络上的实验

为了验证本章提出的稳定策略用在 COPRA 算法中的效果，进行本组实验，将 OCDABSLP 算法与 COPRA 算法进行比较。图4.2∼4.5中的八幅图分别是 OCDABSLP 算法和 COPRA 算法在四组重叠 LFR 基准网络数据集（S7∼S10）上实验结果的 ENMI 和 EQ 指标的对比图。实验中，参数 v 设置为 om 的值。由于 COPRA 算法存在随机性，因此取 10 次实验的平均值作为最后的结果。横轴代表重叠节点所属的社区个数 om，取值从 2 到 8；左侧四幅图的纵轴代表社区划分结果的 ENMI 值，右侧四幅图的纵轴表示实验结果的 EQ 值。

从图4.2∼4.5中可以看出，ODABSLP 算法不仅能够得到稳定的社区发现结果，而且得到的社区结构 ENMI 和 EQ 两个指标都优于 COPRA 算法。验证了本章所提方法在重叠社区发现方面能得到比较好的结果。

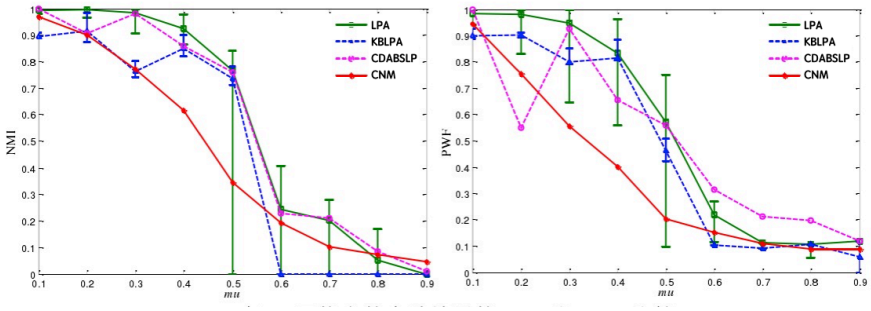

图 4.2　在 S7 网络上的实验结果的 ENMI 和 EQ 比较

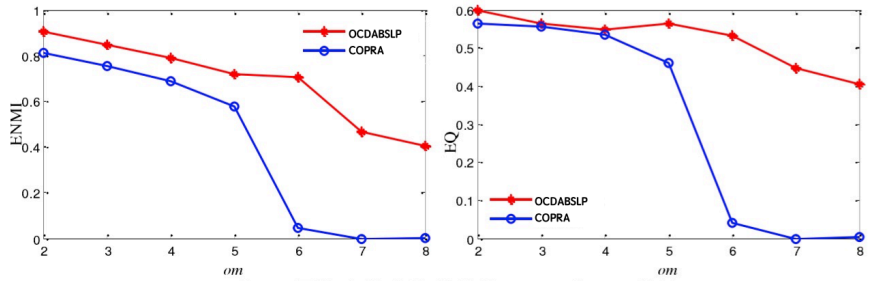

图 4.3　在 S8 网络上的实验结果的 ENMI 和 EQ 比较

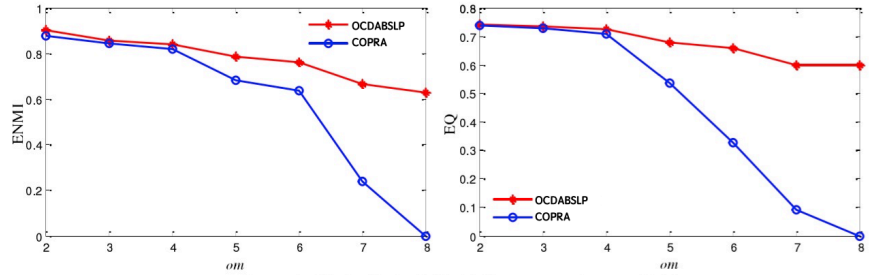

图 4.4　在 S9 网络上的实验结果的 ENMI 和 EQ 比较

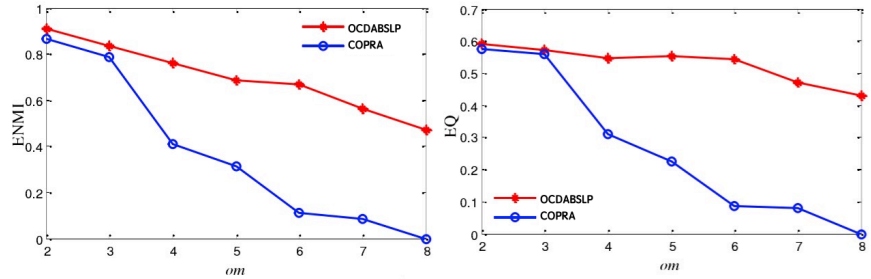

图 4.5　在 S10 网络上的实验结果的 ENMI 和 EQ 比较

4.3.5 实验总结

OCDABSLP 算法采用同步更新策略，在标签更新过程中，当一个节点拥有的所有标签对应的隶属度都小于 1/v，且此时有多个标签的隶属度同时取最大值时，将节点影响值引入到标签隶属度计算公式中，得到这些标签的影响强度，保留影响强度最大的标签，取代传统 COPRA 算法随机保留其中一个标签的方法，提高算法的稳定性。在重叠 LFR 数据集上的实验结果表明 OCDABSLP 算法解决了 COPRA 算法不稳定的问题，能够检测得到较优的重叠社区结构，验证了本章提出的稳定策略在重叠社区发现算法 COPRA 算法中的适用性。

结论

本文采用……。（结论作为学位论文正文的最后部分单独排写，但不加章号。结论是对整个论文主要结果的总结。在结论中应明确指出本研究的创新点，对其应用前景和社会、经济价值等加以预测和评价，并指出今后进一步在本研究方向进行研究工作的展望与设想。结论部分的撰写应简明扼要，突出创新性。）

参考文献

[1] Mcpherson M, Smithlovin L, Cook J M. Birds of a feather: Homophily in social networks[J]. Annual Review of Sociology, 2001, 27(1): 415–444.

[2] Fortunato S. Community detection in graphs[J]. Physics Reports: A Review Section of Physics Letters (Section C), 2010(3/5): 75–174.

[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(12): 7821–7826.

[4] Newman M, Girvan M. Find and evaluating community structure in networks[J]. Physical Review E Statistical Nonlinear Soft Matter Physics, 2004, 69(2 Pt 2): 026113–026113.

[5] Raghavan U N, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks[J]. Physical Review E Statistical Nonlinear Soft Matter Physics, 2007, 76(3 Pt 2): 036106.

[6] Liu S C, Zhu F X, Lin G, et al. A label-propagation-probability-based algorithm for overlapping community detection[J]. Chinese Journal of Computers, 2016.

[7] Ford L R J, Fulkerson D R. Maximal flow through a network[M]. [S.l.]: Birkhäuser Boston, 2009: 243–248.

[8] Pothen A, Simon H D, Liou K P. Partitioning sparse matrices with eigenvectors of graphs[J]. Siam J.matrix Anal.appl, 1990, 11(3): 430–452.

[9] Kernigan R. An efficient heuristic procedure for partitioning graphs[J]. Bell System Technical Journal, 1970, 49.

[10] Leskovec J. Graphs over time: Densification laws, shrinking diameters, explanations and realistic generators[J]. Kdd, 2005: 177–187.

[11] Deng Q, Li Z, Zhang X, et al. Interaction-based social relationship type identification in microblog [M]. [S.l.]: Springer International Publishing, 2013: 151–164.

[12] Natarajan N, Sen P, Chaoji V. Community detection in content-sharing social networks[C]. Ieee/acm International Conference on Advances in Social Networks Analysis and Mining. [S.l.: s.n.], 2013: 82–89.

[13] Abdelbary H A, Elkorany A M, Bahgat R. Utilizing deep learning for content-based community detection[C]. Science and Information Conference. [S.l.: s.n.], 2014: 777–784.

[14] Yin Z, Cao L, Gu Q, et al. Latent community topic analysis:integration of community discovery with topic modeling[J]. Acm Transactions on Intelligent Systems Technology, 2012, 3(4): 1–21.

[15] Sachan M, Contractor D, Faruquie T A, et al. Using content and interactions for discovering communities in social networks[C]. International Conference on World Wide Web. [S.l.: s.n.], 2012: 331–340.

[16] RosenZvi, Michal, Griffiths, et al. The author-topic model for authors and documents[M]. [S.l.: s.n.], 2012: 487–494.

[17] Zhou D, Manavoglu E, Li J, et al. Probabilistic models for discovering e-communities[J]. Proc. 15th Int. Conf. on World Wide Web (WWW'06), 2006: 173–182.

[18] Liu H, Chen H, Lin M, et al. Community detection based on topic distance in social tagging networks [J]. Telkomnika Indonesian Journal of Electrical Engineering, 2014, 12(5).

[19] Peng D, Lei X, Huang T. Dich: A framework for discovering implicit communities hidden in tweets [J]. World Wide Web-internet Web Information Systems, 2015, 18(4): 795–818.

[20] Yang T, Jin R, Chi Y, et al. Combining link and content for community detection: a discriminative approach[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [S.l.: s.n.], 2009: 927–936.

[21] Cohn D, Hofmann T. The missing link: a probabilistic model of document content and hypertext connectivity[C]. International Conference on Neural Information Processing Systems. [S.l.: s.n.], 2001: 409–415.

[22] Getoor L, Friedman N, Koller D, et al. Learning probabilistic models of link structure[J]. Journal of Machine Learning Research, 2003, 3(4): 679–707.

[23] Xie J, Kelley S, Szymanski B K. Overlapping community detection in networks[M]. [S.l.: s.n.], 2013.

[24] Palla G, Dere Nyi I, Farkas I S, et al. Uncovering the overlapping community structure[M]. [S.l.: s.n.], 2005.

[25] Lancichinetti A, Fortunato S, Kertész J. Detecting the overlapping and hierarchical community structure of complex networks[J]. New Journal of Physics, 2012, 11(3): 19–44.

[26] Becker E, Robisson B, Chapple C E, et al. Multifunctional proteins revealed by overlapping clustering in protein interaction network[J]. Bioinformatics, 2012, 28(1): 84–90.

[27] Magdon-Ismail M, Purnell J. Ssde-cluster: Fast overlapping clustering of networks using sampled spectral distance embedding and gmms[C]. IEEE Third International Conference on Privacy, Security, Risk and Trust. [S.l.: s.n.], 2010: 756–759.

[28] Lei X, Wu S, Ge L, et al. Clustering and overlapping modules detection in ppi network based on ibfo [J]. Proteomics, 2013, 13(2): 278–290.

[29] Ren W, Yan G, Liao X. A simple probabilistic algorithm for detecting community structure in social networks[J]. Physics, 2007: 36–40.

[30] Erosheva E, Fienberg S, Lafferty J. Mixed-membership models of scientific publications[J]. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101 Suppl 1 (1): 5220.

[31] Nallapati R M, Ahmed A, Xing E P, et al. Joint latent topic models for text and citations[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, Usa, August. [S.l.: s.n.], 2008: 542–550.

[32] Dietz L, Bickel S, Scheffer T. Unsupervised prediction of citation influences[C]. International Conference on Machine Learning. [S.l.: s.n.], 2007: 233–240.

[33] Gruber A, Rosenzvi M, Weiss Y. Latent topic models for hypertext[J]. Uai, 2008.

[34] Zhu S, Yu K, Chi Y, et al. Combining content and link for classification[M]. [S.l.: s.n.], 2007: 487–494.

[35] Yu S, Moor B D, Moreau Y. Clustering by heterogenous data fusion: framework and applications[J]. Dec-2008, 2008.

[36] DJ W, SH S. Collectivedynamics of ’small-world’networks[C]. Nature. [S.l.: s.n.], 1998: 440–442.

[37] Leung I X Y, Hui P, Liò P, et al. Towards real-time community detection in large networks[J]. Physical Review E Statistical Nonlinear Soft Matter Physics, 2009, 79(6 Pt 2): 066107.

[38] He M, Leng M, Li F, et al. A node importance based label propagation approach for community detection[C]. the seventh international conference on intelligent system and knowledge engineering, iske 2012/the 1st international conference on cognitive system and information processing, csip 2012. [S.l.: s.n.], 2014.

[39] Xie J, Szymanski B K. Labelrank: A stabilized label propagation algorithm for community detection in networks[J]. ., 2013: 138–143.

致谢

光阴荏苒，岁月如梭，两年的研究生生活转瞬即逝。值此毕业论文即将完稿之际，我对帮助过我的老师、同学以及亲友表达由衷的感谢，并对本硕共培养了我 6 年的母校北京理工大学致以诚挚的敬意。

首先感谢我的两位导师：张欣老师和金福生老师。张老师是我名义上的导师，但是实际上两年研究生生涯我接触更多的是金老师。在参与金老师负责的实验室所承接的项目的工作中，我积累到了宝贵的项目经验。本论文的工作也包含了金老师悉心的监督和指导，在论文的撰写上提出了很多宝贵意见。感谢两位导师的帮助和指导。

感谢陪伴我两年的舍友张俊逸、谢辰和刘哲湘。两年的朝夕相处与你们建立了深厚的情谊，不论是学习还是生活中都少不了你们的帮助和支持。看着你们如今都找到满意的工作，有了很好的归宿，真心替你们开心，这一毕业就是各奔东西了，祝大家都前程似锦吧。

感谢我的同窗们李璟明、蔡天倚、王宇侠等，感谢帮助过我的学长学姐龚思胜、朱冲冲、孙晨光等，感谢你们对我学习生活上的帮助和支持。

感谢我的父母和家人，是他们无私的关怀和奉献支撑我完成了学业。

最后，感谢各位参加论文评审和论文答辩的老师们的批评与指导！