燕山大学
YANSHAN UNIVERSITY

硕 士 学 位 论 文

MASTER'S DISSERTATION

论文题目 基于 k-shell 的复杂网络影响力最大化算法研究

作者姓名 曹迪

学科专业 计算机科学与技术

指导教师 刘永山教授

2017 年 5 月

中图分类号：TP392　　学校代码：10216
UDC：004.65　　密级：公开

工学硕士学位论文

基于 k-shell 的复杂网络影响力最大化算法研究

硕士研究生：曹迪
导　　师：刘永山教授
申请学位：工学硕士
学科专业：计算机科学与技术
所在单位：信息科学与工程学院
答辩日期：2017 年 5 月
授予学位单位：燕山大学

A Dissertation in Computer Science and Technology

RESEARCH ON THE ALGORITHM OF MAXIMIZING INFLUENCE OF COMPLEX NETWORK BASED ON K-SHELL

By Cao Di

Supervisor:Professor Liu Yongshan

Yanshan University

May, 2017

燕山大学硕士学位论文原创性声明

本人郑重声明：此处所提交的硕士学位论文《基于 k-shell 的复杂网络影响力最大化算法研究》，是本人在导师指导下，在燕山大学攻读硕士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签字：　　日期：　　年　月　日

燕山大学硕士学位论文使用授权书

《基于 k-shell 的复杂网络影响力最大化算法研究》系本人在燕山大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归燕山大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解燕山大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅。本人授权燕山大学，可以采用影印、缩印或其它复制手段保存论文，可以公布论文的全部或部分内容。

保密□，在　　年解密后适用本授权书。

本学位论文属于

不保密□。

(请在以上相应方框内打“√”)

作者签名：　　日期：　　年　月　日

导师签名：　　日期：　　年　月　日

摘 要

近年来，影响力最大化问题是复杂网络研究领域的一个具有重大实际意义的课题。该课题概括为：寻找网络中规模为 k 的具有最大影响范围的节点集。k-shell 是图论中的重要概念。实验表明，k-shell 算法能够有效的识别出网络的核心。但是该算法只是粗糙的衡量了节点在网络中的位置，并未对每一个节点做更细致的影响力分析，因而影响了种子节点集的质量并造成了影响范围的不稳定性。

本文通过阅读大量的文献资料，学习了复杂网络领域的相关基础理论并深入分析了 k-shell 算法和网络的社区结构特性。结合目前的研究现状及存在的问题，提出了以下两种算法。

首先，本文针对 k-shell 算法划分粗糙的问题提出了一种基于 k-shell 的影响力最大化算法。考虑处于同一 k-shell 层节点的局部信息，定义节点的影响力。并对种子节点集进行优化，采用局部能量缩减策略来降低种子节点影响力的重叠。

其次，深入分析网络的社区结构特性，提出一种基于社区结构的影响力最大化算法。首先采用 Louvain Methord 算法对网络进行社区划分，对每个社区，利用 k-shell 算法找出核心节点集。其次，根据社区间连边的数量以及所连接社区的规模来定义连边的权重，并根据每个边界节点拥有社区间连边数目及连边的权重来定义边界节点的枢纽重要度。最后，根据计算的比例从每个社区中选择部分核心节点和从边界节点集中选择若干边界节点共同组成种子节点集。

最后，选取了 4 个真实的网络数据集在独立级联模型上进行仿真实验，并与经典的中心性算法和 k-shell 算法进行对比，分析实验结果并得出结论。

关键词：复杂网络；k-shell 算法；局部信息；社区结构

Abstract

In recent years,the problem of maximizing influence is a significant topic in the complex network research.The topic is summarized as follows:finding a set of nodes with a size of k which has the greatest impact on the whole network.The k-shell is an important concept in graph theory.Experiments show that the k-shell algorithm can effectively identify the core of the network.However,the algorithm only measures the position of nodes in the network,and does not do more detailed analysis of each node.So the quality of the seed node set is bad and the influence range is unstable.

In this paper,by reading a lot of literature,we study the basic theory of complex network and deeply analyze the k-shell algorithm and the network structure characteristics of the network.Combining with the current research status and existing problems,the following two algorithms are proposed.

First of all,this paper proposes a maximum algorithm based on k-shell for the rough problem of k-shell algorithm.Consider the local information of the node which is at the same k-shell layer to define its influence.And optimize the seed node set by the local energy attenuation strategy.

Secondly,we deeply analyze the community structure characteristics of the network,and propose an algorithm to maximize the influence which is based on community structure.First,useing the Louvain Methord algorithm to classify the network into several communities.For each community,we use the k-shell algorithm to find the core node set.Second,defining the weight of the edge according to the number of links between communities and the size of the connected communities.And defining the hub importance of the boundary nodes according to the number of inter-community connections and the weights of each boundary node. Finally,a subset of the nodes are selected from each community according to the calculated ratio and a number of boundary nodes are selected from the boundary node set to form a seed node set.

Finally,we selected four real network datasets to simulate the experiment on the

independent cascade model,and compared with the classical central algorithm and k-shell algorithm, and analyze the experimental results to make a conclusion.

Keywords: complex network;k-shell;location information;community

目 录

第 1 章 绪 论

1.1 研究背景和意义

20 世纪 60 年代初，美国著名社会心理学家米尔格伦提出了六度理论。该理论的观点是[1]：“仅仅需要五个人作为中介，你就可以认识地球上的任何人”这句话侧面的说明了世界是如此之小，人与人之间具有高度的相关性。六度理论是复杂性网络的最早起源。复杂网络由网络中每个成员及它们之间的关系构成，成员表示节点，关系表示边。

复杂网络在本文的生活中广泛存在，许多领域的复杂问题和关系都可以抽象成复杂网络。比如人类的社会关系网络[2]，节点是社会中的每个人，边是指人们之间的相识关系；人们每天登陆的互联网，节点则是网络中的每台电脑，边则是电脑与电脑之间的网络连接。常见的复杂网络还有新陈代谢网络[3]、交通网络[4]、食物链网络[5]、电力网络[6]、神经网络[7]、以及学术研究中的科学家合作网络[8]等。在社会学、生命科学、自然科学、计算机科学及其他技术领域也都涉及到复杂网络的相关研究。因此，复杂网络作为一种描述现实中各种关系的结构，亟待着人类对其进行更深入的研究。

病毒式营销，是目前网络上一种常见的低成本、低耗时、高效率的宣传方法。其利用的是人们口碑传播[9]的原理。在“病毒式营销”中，通常商家为了推广某种产品，会先将这种产品免费提供给部分消费者进行试用，如果消费者认可了这种产品便会推荐给他的朋友，而他的朋友也会重复这个行为将产品推广出去。利用这种“口碑传播”，产品的信息就可以以快如闪电的速度被广大消费者所熟知。当今时代，互联网技术发展迅猛，国内外出现了许多的网络社交平台，例如 Facebook、Twitter、微信、微博。这些网络社交平台拓宽了人们的朋友圈，增进了人与人之间的信息交流，也为病毒式营销提供了更有效的传播媒介。通过互联网，这用营销手段获得了人们的高度认可[10]。在“病毒式营销”中，要想产品信息被人们耳熟能详并且高度认可，初始目标人群的选择非常重要。初始传播者应该是那些众朋广友、声名显赫、在人群中有影响力的人。这是影响力最大化问题产生的最初背景。19 世纪 20 年代初，那时国外学者开始关注到这个课题并经过多年的努力研究取得了很多成果，并形成了相对完善的理论体系。现在这个领域

的研究主要包括：社区发现算法、社区可视化和影响力最大化等。

近年来，很多国内学者投身于复杂网络影响力最大化问题的研究中。影响力最大化问题被引入到各个社会网络中去解决众多领域的科学难题。例如在罪犯关系网络[11]中，通过分析网络结构找出犯罪团伙的头目和各个犯罪分子之间的关系，从而制定抓捕策略，将犯罪分子绳子以法；在电网中，为防止网络中个别设备的故障而导致大面积的电力瘫痪[12]，需要找到网络中的核心设备并保护起来；在网页搜索方面，需要在数以亿计的网页中找到可信度最高的网页并返回给上网者；在谣言的抑制领域中，找到最大的谣言传播者并切断最重要的传播路径以抑制谣言的传播。因此，如何找到网络中的关键节点已成为一项至关重要的工作。

综上所述，影响力最大化问题的研究，具有重要的理论意义和实践价值，能够带动众多相关领域及技术的发展，从而能够以更低的成本和更高的效率去解决更多的实际问题，具有非常广阔的前景。

1.2 国内外研究现状

影响力最大化问题是近年来复杂网络领域研究的热点问题，国内外学者们经过多年的研究，以不同的思维角度相继提出了众多该问题的解决方法。

Kemple 和 Kleinberg 提出了影响范围达到最优解 63%的贪心算法[13]。该算法把每个节点都作为一次种子节点去计算激活增量，然后选择激活增量大的节点加入到种子节点集，时间复杂度高，不适用于大型网络。后续很多研究者对 KK 算法做出了改进，Leskovec 等人提出了一种简化贪心算法的 CELF 算法[14]。田家堂等提出了基于 LT 模型积累特性的启发式算法[15]。Chen 等提出了简化网络结构的 New-greedy 算法和 Mix-greedy 算法[16,17]。以上算法较贪心算法的复杂度都有所降低，但对大型网络仍不适用。

几种传统的中心性算法：度中心性、紧密中心性、介数中心性。这些算法都是从网络的拓扑结构角度来定义节点的影响力。一般认为，度中心性反映的是节点一步之内的影响力[18]。一般认为节点的度数越大，直接受影响的人越多。紧密中心性和介数中心性虽然有不错的影响范围，但是是基于整个网络最短路径的算法，复杂度较高。L C Freeman 等人将介数扩展到了流介数，认为网络中所有节点对之间的路径是等价的，不应只考虑节点对之间最短路径[19]。Chen 等人考虑了节点四步可到达的邻居的信息，提出一种考虑半局部结构的算法，综合结果优于介

数算法，并且复杂度较低[20]。另外，还有一些学者考虑通过破坏网络中的某些节点或者连接而造成的损失来确定节点的重要性。陈勇等人采用删除节点及连接法，认为从图中去掉节点以及相关联的链路后，所得到的图对应的生成树数目越少，则该组节点越重要[21]。另外，Restrepo 定义了一个描述动态网络节点和链接的重要性影响的最大特征值，然后移除网络中的每个节点，通过比较特征值的变化来确定该节点在网路中的重要性[22]。

PageRank 算法是著名的网页排序算法，它通过衡量其他网页到该网页的链接数来确定网页的重要性。近年来，很多学者对 PageRank 算法提出了改进和优化。刘耀庭等深入分析了社交网路中的人际关系，认为社交网络中的人际结构决定了用户的影响力，通过引入边的概率方向将无相关系进行定向，提出了基于人际结构的 UserRank 算法[23]。Cha M 搜集了大量的 Twitter 数据并通过观察提出了三个 Twitter 网络的影响力度量指标：入度、转发、提及[24]。Weng 等充分挖掘了 Twitter 平台的数据信息，考虑了用户之间的局部相似性和链路结构，提出一种针对于 Twitter 网站的 TwitterRank 算法[25]。实验表明，该算法优于 PageRank 算法。

Kitsak 等人提出用 k-壳分解法(k-shell decomposition)来度量节点的影响力[26]。k-shell 算法将网络分为若干层，能够较合理的给出每个节点在网络中的位置。但 k-shell 算法只是一种粗粒度的划分网络的方法。Liu J G 等人提出了一种基于 k-shell 的改进算法，该算法考虑了目标节点与 k-shell 划分的最内层节点的距离，认为到最内层所有节点的距离和最小的节点的影响力大[27]。该算法的弊端就是容易造成种子节点堆聚现象，进而造成影响力重叠。顾亦然等提出 KSA 算法，该算法在考虑节点重要度(即 k-shell 值)的基础上，同时考虑邻居节点的重要度贡献[28]。Joonhyun Bae 等考虑了节点的所有二级邻居的核数(kshell 值)提出了一种基于局部信息的方法[29]。这两种算法都仅仅考虑的邻居节点的核数，不够全面。Zeng 等人认为 k-shell 分解算法仅考虑了节点的剩余度而完全忽视了节点的消耗度，因此提出一种基于混合度的 k-shell 分解的改进算法，并通过仿真实验证明该算法优于 k-shell 算法和度中心性算法[30]。Hu 等人提出一种考虑社区结构上的 k-shell 算法，在 SIR 模型上的仿真实验表明该方法优于 Kitsak 等人的方法[31]。

社区结构是复杂网络所普遍具有的一个拓扑特性[32]。近年来，很多研究者都把影响力最大化问题放到社区中去研究。其中，最早提出这种想法的是 Galstyan 等人，但是他所研究的只局限于两个连接松散的社区，并不具有代表性[33]。冀进

朝等提出一种基于社区结构的 AMICS 方法，该方法轮换选择跨越社区数最多的 k 个节点作为种子节点[34]。虽然，该算法重点考虑了社区间枢纽节点的作用，但忽略了社区内部核心节点的地位。CaO 等提出一种基于社区结构的动态资源分配算法，该算法在已划分好社区结构的网络上，提出把 k 个种子节点分别分配到各个社区中去做仿真实验，最后累加各个社区的影响范围得到整个网络的影响范围[35]。该算法只重视了社区内部节点的重要性，忽视了社区之间的联系，与 AMICS 算法思想恰好相反。郭进时等提出一种综合考虑社区内部节点和社区间枢纽节点的影响力最大化算法，该算法的一点不足就是在选取每个社区的种子节点数时只考虑了社区的规模，不够全面，例如还可以考虑该社区被其他社区激活的能力[36]。

综上分析，国内外学者虽然给出了很多优秀的算法，但还有一些不足之处。k-shell 算法划分过于粗糙，无法区分同一层节点的影响力。对 k-shell 的各种改进算法都只考虑了单个节点的影响力，没有对规模为 k 的种子集进行整体性的考虑。另外，一些基于社区结构的算法只考虑了社区内核心节点或只考虑了社区间枢纽节点，还有一些基于社区结构的算法虽然综合考虑了这两类节点，但在每个社区种子节点的选取比例上考虑的不够全面，仅仅以社区的规模作为依据，没有考虑到社区间的互相激活。针对以上基于 k-shell 算法和基于社区结构算法的不足，本文提出了两种改进的算法。在下面的第 3 章和第 4 章给出详细的介绍。

1.3 本文的研究内容

现实生产生活的很多实体之间的复杂关系都可以抽象成网络，通过分析网络的拓扑特点，找出网络结构中的重要节点，能够帮助本文精准的找出问题，省时省力有效的解决问题。这就是影响力最大化问题的现实意义。本文在分析大量优秀文献的基础上，对前人的算法取其精华，去其糟粕。提出了两种算法，本文主要做了以下几方面的工作：

查阅相关资料，了解影响力最大化问题的理论背景及研究意义。细读该领域内的优秀图书及文献资料，了解目前国内外该问题的研究现状及经典算法。

学习复杂网络的基础理论知识及仿真模型，分析与该课题相关的经典算法的优缺点。

重点分析了 k-shell 算法的优缺点，并针对其缺点即划分粗糙，无法区分同一层节点的影响力提出了一种改进算法，综合考虑了节点在网络中的位置及其邻居

信息。并从种子节点集整体的角度考虑，提出了一种基于 k-shell 局部结构的能量缩减算法。并在一个小型的人工网络上进行分析对比。

针对复杂网络规模越来越大、结构越来越复杂的现实问题，考虑分治的思想—将大型网络划分成若干小网络，分而治之。本文分析了复杂网络的社区结构特性及经典的社区划分算法。提出一种基于社区结构的影响力最大化算法，定义社区间连边的权值。从社区核心节点集和边界节点集两方面考虑影响力最大化问题。

在独立级联模型上对复杂网络真实数据集进行仿真实验，分析实验结果得出结论。

1.4 本文的组织结构

从整体结构上看，本文共分为 5 章。

第 2 章为复杂网络的基础理论，主要介绍了网络及复杂网络的定义及图论表示、三种主要的网络的统计特征量：度分布、平均路径长度和聚类系数。然后介绍了复杂网络的几种典型的中心性算法主要包括：度中心性、紧密中心性、介数中心性和特征向量中心性。

第 3 章首先详细介绍了 k-shell 分解算法并在一个小型人工网络上分析该算法的优缺点，并针对其不足给出了一种改进算法。该算法综合考虑了节点的位置和节点的邻居信息来定义单个节点的 KSLC 影响力。然后考虑传播过程中的多种因素并采用局部能量缩减的思想选出规模为 k 的种子节点集，提出一种基于 k-shell 局部结构的能量缩减算法。并在小型人工网络上进行分析对比，得出算法的有效性。

第 4 章首先介绍了复杂网络的社区结构特性，接着介绍了本文采用的 Louvain Method 社区划分算法的算法思想和原理。最后，提出了一种基于社区结构的影响力最大化算法。在该算法中，种子节点集包括社区内核心节点和整个网络的边界节点两部分。通过定义有效连边、社区联系度来定义社区间连边的权重。最后综合边界节点到其他社区连边的条数、连边的权值定义边界节点的枢纽重要性。并定义社区激活度来确定每个社区选择种子节点的比例。

第 5 章为实验部分，详细介绍了实验环境、数据集和仿真模型的实现。并将第 3 章和第 4 章提出的两种算法分别进行仿真实验，对比分析实验结果，以证明其有效性。

第 2 章 相关理论基础

本章主要介绍复杂网络的一些基础的理论，主要包括网络及复杂网络的定义及图论表示，网络的一些基本的统计特征量以及几种影响力最大化问题的经典算法和该研究领域一些常用的经典的信息传播模型。

2.1 网络的定义及图论表示

统计物理学角度：网络是由许多个体和这些个体间的相互作用组成的系统。

图论的角度：网络可以用图 $G=(V,E)$ 来描述，其中 $V=\{1,2,\ldots,N\}$ 并且 $V\neq\phi$ 为节点集，$E\in V\times V$ 为边的集合。边集中的每一条边都与非空节点集中的一对节点相对应，其中节点是网络的基本元素，边是元素间的相互作用或关系。图 2-1 为新西兰 62 只宽吻海豚的关系网络图。

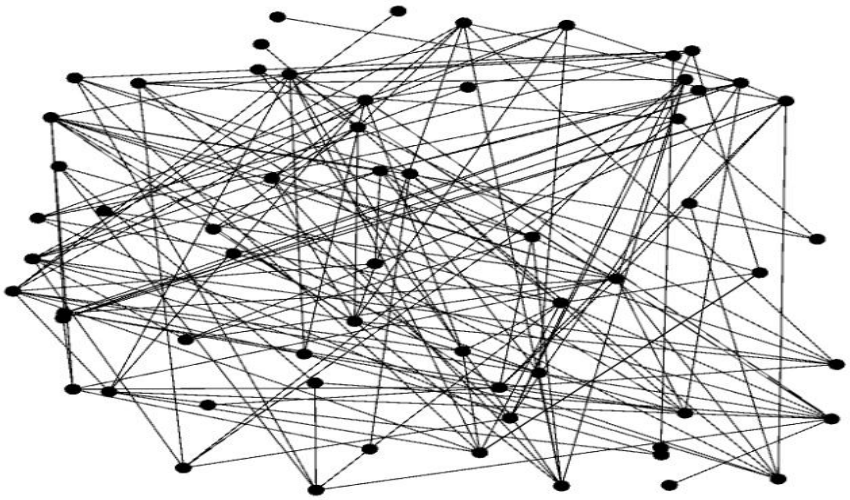


图 2-1 海豚关系网络图

图与网络是一一对应的。每一个网络都唯一确定的对应着一个图。对图中的任意边*(u,v)*，如果边*(u,v)*与边*(u,v)*是一条边，即无需区分边的方向和起始点，这样的边叫做无向边，反之叫做有向边。另外，在有些网络中，每条边会有一个度量值。该度量值可以表示边的长短、边的重要性、通过这条边的难易程度等，这个度量值叫做边的权值。

本文根据图的边是否有方向和是否有权重将图分为以下四种类型：

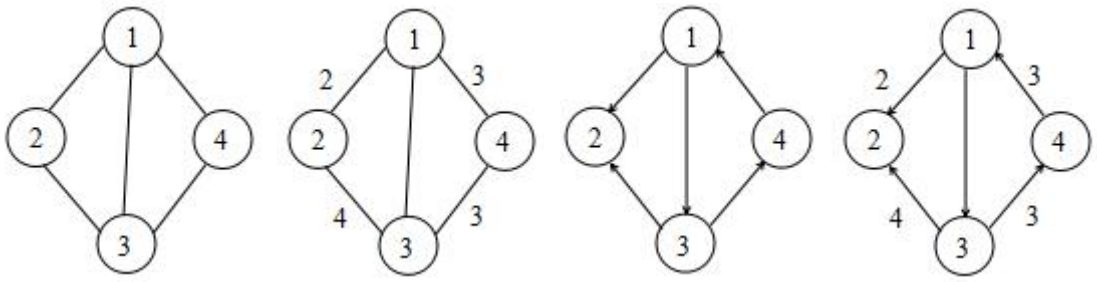
无向无权图：图中的边无需区分起始点也无度量值。

无向加权图：图中的边无需区分起始点但有度量值。

有向无权图：图中的边有起始点但无度量值。

有向加权图：图中的边有起始点也有度量值。

四种类型的网络图如图 2-2 所示。



a)无向无权图　　b)无向加权图　　c)有向无权图　　d)有向加权图

图 2-2 四种类型的网络图

复杂网络简而言之就是结构和连接呈现高度复杂性和无规律性的网络。主要包括：无标度网络、随机网络、小世界网络。其复杂性主要是指。

(1) 节点数目多。现实中某些复杂系统中的节点已达到数亿。节点象征含义的多样性，可以代表自然界的一切事物，例如人、网页、计算机、神经元等。

(2) 网络是动态变化的。例如万维网，每分钟甚至每秒钟，都会有新的网页被打开，旧的网页被关闭，从而导致网络结构不断变化。

(3) 连边的多样性。由于边可以有方向和意义，还可以给定一个度量的权值，因此处理起来比较复杂。例如科学家合著论文网络中，边的权值代表两科学家共同合作的论文的数量。

2.2 网络的统计特征量

在复杂网络的研究中，现实世界中的实体及其关系抽象成的网络往往规模巨大，关系复杂。需要通过定义一些指标来观察网络结构的拓扑性质，获得一个对网络整体的简单直观的了解。因此，常常结合统计学的理论来进行研究。下面本文介绍近年来复杂网络科学研究中关注最多的 3 个拓扑性质：度分布、平均路径长度和聚类系数。

2.2.1 度分布

复杂网络的度分布(Degree distribution)是从概率统计的角度来定义的，记为

P(k)。无向网中 P(k)指的是网络中度值为 k 的节点占整个网络的比重。有向网络中，有出度与入度之分，因此就对应有出度分布(Out-degree distribution)和入度分布(In-degree distribution)记为 $P(k^{out})$ 和 $P(k^{in})$。相应可得，$P(k^{out})$ 是指网络中出度为 k 的节点占整个网络的比重。$P(k^{in})$ 是指网络中入度为 k 的节点的占整个网络的比重。

网络的度分布反映了网络节点间连接稀疏情况，常见的度分布有泊松分布，幂律分布。

泊松分布满足的关系式见式(2-1)。

$$P(k)=\frac{\lambda^k e^{-\lambda}}{k!}(\lambda>0) \tag{2-1}$$

幂律分布满足的关系式见式(2-2)。

$$P(k)\sim k^{-r} \tag{2-2}$$

其中($\gamma>0$ 为幂指数，且 $2\le\gamma\le3$)。

2.2.2 平均路径长度

平均路径长度是将网络中所有节点对的最小距离加到一起，再除以该图对应的完全图中的边数。平均路径长度计算方法见式(2-3)。

$$L=\frac{1}{\frac{1}{2}N(N-1)}\sum_{i\ge j}d_{ij} \tag{2-3}$$

式中 N 为网络的节点数。d_{ij} 表示节点 i 和节点 j 之间的最小距离。在复杂网络的影响力最大化研究中，网络的平均路径长度可以侧面的反映出消息在此网络中传播速度。一般来说，平均路径长度短的网络节点对之间路径较短，因此信息传播的速度会更快。

2.2.3 聚类系数

聚类系数包括节点的聚类系数和网络的聚类系数。节点的聚类系数表示节点的邻居连接的密切程度。例如，将公司内的所有员工及他们之间合作的关系抽象成网络，当然老板和所有的员工都是合作的关系。对于老板这个节点，它的聚类系数越高，说明他的员工之间的团结性和凝聚力越好。

节点 v 聚类系数反映了以节点 i 为中心近似于一个团的连接的紧密程度。

网络中节点 i 的聚类系数定义见式(2-4)。

$$C_i = \frac{E_i}{(k_i(k_i-1))/2} = \frac{2E_i}{k_i(k_i-1)} \tag{2-4}$$

这里，k_i 指的是节点 i 的度，E_i 指的是节点 i 的 k_i 个邻居节点之间连边数。从几何的角度看，E_i 是指以 i 为顶点的三角形的个数。因此，节点 i 的聚类系数还可以定义为式(2-5)。

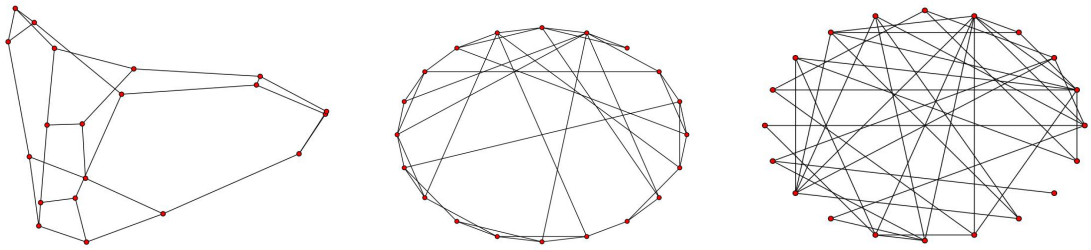
$$C_i = \frac{\text{包含节点}i\text{的三角形的数目}}{\text{以节点}i\text{为中心的连通三元组的数目}} \tag{2-5}$$

在影响力最大化的研究中，若某个节点 i 的聚类系数比较高，那么节点 i 的邻居节点被激活成功的比例就大。因为若节点 i 有 5 个邻居节点，分别为 j、k、l、m、n，节点 i 只激活了 j 和 k，由于 j、k、l、m、n 之间连接的较紧密，所以 l、m、n 极有可能被 j 和 k 激活成功。另外聚类系数较高的节点邻居节点之间影响力交叉重叠现象较严重，所以也应尽量少选择互为邻居的节点作为种子节点。

2.3 复杂网络的基本特性

2.3.1 小世界特性

小世界网络是真实世界网络的一个近似模拟，它反映出的是世界如此之小以及人们之间的高度联系。小世界网络和真实世界的网络有很多的相似点，例如较小的最短路径长度和较大的聚类系数。WS 网络是通过随机的改变规则网络中某些连边而形成的。改变连边的方法如下：对于一个规则的环形最近邻耦合网络，对其所有的边，保留一个端点，让另一个端点以概率 p 随机连到其他端点上，并保证能出现两个节点的环，也不能出现自身的环。在 WS 模型中，p 取 0 时得到的是规则网络，p 取 1 时得到的是随机网络，p 取 0 到 1 之间时得到的是 WS 网络。通过调节 p 值可以得到这三种网络，如图 2-3 所示。



a)规则网络 p=0　　b)WS 网络 0<p<1　　c)随机网络 p=1

图 2-3 三种网络模型

2.3.2 无标度特性

1998 年，Barabasi 与 Albert 等人借助机器人爬行观察万维网(World Wide Web)的直径时发现了无标度网络。他们发现，World Wide Web 仅仅由少量多连接的页面串成。超过 8 成的网页的连接数不足 4 个，但却竟有不到万分之一的页面的连接数超过 1000。于是在 1999 年，Barabasi 与 Albert 提出了著名的 BA 模型[37,38]，给出无标度网络的定义：度分布具有幂律尾部特征的网络。

BA 模型具有以下几个特性：

(1) BA 网络具有幂律度分布，见式(2-6)。

$$P(k) \approx 2m^2 k^{-\gamma}, k \geq m, \gamma = 3 \tag{2-6}$$

其中γ称为度指数，并且 m 与γ互相独立。

(2) 平均路径短，平均路径长度满足的关系见式(2-7)。

$$l \sim \ln(N) / \ln\ln(N) \tag{2-7}$$

(3) 集群系数小，集群系数满足的关系见式(2-8)。

$$C \sim N^{-0.75} \tag{2-8}$$

2.4 影响力最大化问题的经典算法

中心性指标是目前评价节点影响力时较为常用的指标，主要包括度中心性(Degree Centrity)、紧密中心性(Closeness Centrity)、介数中心性(Betweeness Centrity)和特征向量中心性(Eigenvector Centrality)。

(1) 度中心性

度中心性指标是最简单的衡量节点影响力的指标。它只考虑节点一阶邻居的多少，没有对节点的影响力进行更长远更全面的估计。在无向网络中，度只和与节点直接相连的节点数目有关。在有向图中，度分出度和入度两种，节点 i 的出度 k^{out} 指的是以 i 为起始点的有向边的数目，节点 i 的入度 k^{in} 是指以 i 为终点的有向边的数目。在社交网络中，入度代表着一个人的受欢迎程度，出度代表着一个人的合群度。

在加权图中，节点的度等于与节点相连的边的权值的总和。为了比较不同规模的网络中具有相同度值的节点的影响力大小，定义节点 v_i 的归一化度中心性指标，见式(2-9)。

$$DC(i)=\frac{k_i}{n-1}(n=|V|) \tag{2-9}$$

实验表明，在很多网络中，度中心性指标简单、计算复杂度低，并且效果也还不错。

(2) 紧密中心性

紧密中心性考虑的是节点的平均传播距离，认为距离所有其他节点越近的节点越能将消息快速的传播出去。定义任意一个节点 i 到网络中所有其他节点的平均最短距离为节点 i 与其他所有节点构成的节点对的最小距离和与理想情况下的距离和 n-1 的比值，具体定义见式(2-10)。

$$d_i=\frac{1}{n-1}\sum_{j\neq i}d_{ij} \tag{2-10}$$

d_i 越小节点 i 到其他所有节点距离越近，则节点 i 仅需要短短几步就能将消息告知所有人。因此紧密中心性应该与 d_i 成反比，满足的关系见式(2-11)。

$$CC_i=\frac{1}{d_i}=\frac{n-1}{\sum_{j\neq i}d_{ij}} \tag{2-11}$$

通过分析式(2-11)可知，紧密中心性衡量的是节点将信息传递给其他人的速度。公式(2-11)需要先计算每个节点到其他所有节点的最小距离，常用的是 floyd 算法，复杂度为 $O(n^3)$。可见，对于大型网路，紧密中心性算法的复杂度较高。

(3) 介数中心性

介数中心性衡量的是某个节点 i 在网络所有最短路径中的枢纽地位，可以近似比喻为现实生活中的铁路枢纽站。节点 i 的介数定义为在网络中所有不以 i 为端点的节点对中，最短路径中经过节点 i 的节点对的比例。其计算方法见式(2-12)。

$$BC(i)=\sum_{i\neq s,i\neq t,s\neq t}\frac{g_{st}^{i}}{g_{st}} \tag{2-12}$$

式中 g_{st} 为网络中不以 i 为端点的总的节点对的数目，g_{st}^{i} 为最短路径中经过节点 i 的节点对数目。

在研究网络的影响力最大化问题中，介数中心性得到的节点不一定是传播能力最强的节点，但却是信息传播中所不可缺少的节点。因为该节点的缺失会导致很多条最短路径的中断，该算法可以说是从破坏力的角度来衡量节点的重要性。

为了在不同类型的网络中研究介数，定义一个归一化的介数见式(2-13)。

$$BC'(i)=\frac{2}{(n-1)(n-2)}\sum\frac{g_{st}^{i}}{g_{st}} \tag{2-13}$$

介数中心性的实现也是以最短路径算法为基础，复杂度为$O(n^3)$，复杂度较高，不适用于大型网络。

(4) 特征向量中心性

特征向量中心性在衡量节点 i 的影响力时，综合考虑了节点 i 的邻居节点的数目和邻居节点的重要性。特征向量值较高的节点既有可能和大量影响力一般的节点相连，也有可能和少量的影响力很大的节点相连。记 X_i 为节点 i 的影响力，其计算方法见式(2-14)。

$$EC(i)=x_i=c\sum^{n}a_{ij}x_j \tag{2-14}$$

式中 c 为一常数，将整个网络所有节点的影响力写成行向量的形式见式(2-15)

$$x=[x_1,x_2,x_3,\ldots,x_n]^T \tag{2-15}$$

因此可以将整个网络所有节点的影响力简化成公式(2-16)

$$x=cAx \tag{2-16}$$

式中 A 为网络的邻接矩阵。该算法的思想就是节点 i 的影响力由 i 的邻居节点 j 的数目和影响力决定，而邻居节点 j 的影响力又由 j 的邻居节点 u 决定。这样一层一层的迭代下去，文献[34]证明，每一步迭代过程中，如果用 x 除以邻接矩阵 A 的主特征值，上述方程就会得到一个收敛的非零解即 $x=\lambda^{-1}Ax$。

当网络中存在若干个度值非常大的节点时，该算法容易造成影响力集聚在度值大的节点上。为了改进算法的这一弊端，Martin 等人提出一种基于 non backtracking 矩阵的替代中心性算法，有效避免了特征向量中心性算法的弊端并且获得了更优的实验结果[39]。

(5) 算法总结

研究复杂网络中影响力最大化问题的算法有很多种，但从大体的类别上来说，可以分为两种主要的求解思想：(1) 正面求解思想。分析网络的结构、节点的特征，使所选的节点能够将消息广泛的传播出去，例如度中心性算法、介数中心性算法、紧密中心性算法、特征向量中心性等。(2) 基于破坏的思想。通过衡量破坏网络结构、破坏节点或连接所造成的损失来评价节点的影响力。例如节点删除最短路径法、节点收缩法等。

2.5 影响力传播模型

2.5.1 传播机制

影响力传播模型定义了影响力在网络上的传播机制，是研究影响力最大化问题不可缺少的实验平台，主要的有：独立级联模型(Independent Cascade，IC)和线性阈值模型(Linear Threshold，LT)和 SIR 传染病模型，下面对这三个模型进行一下说明：

(1) 初始时刻，网络 G 上的节点只有两种状态：激活(active)状态和未激活(inactive)。在有向图中，只有状态为激活的节点对它的终点才具有影响力，状态为未激活的节点对它的终点没有作用。而在无向图中，因为无需区分起始点，所以处于激活状态的节点对它的邻居都有影响力。当一个节点被其他节点成功影响时，称此节点被激活。

(2) 每个处于激活状态的节点对它的所有邻居节点(有向图中指出边邻居)都影响一次(无须考虑是否影响成功)后，该节点仍保持激活状态，但已不具有影响力，不能够再去影响其他节点。

(3) 在有向图里，影响是单向的，只能始点对终点，不能倒置。而在无向图中，不区分始点和终点。边的两端点彼此之间都能互相影响。

2.5.2 独立级联模型

独立级联模型是在概率论和交互离子系统(interacting particle system,IPS)的基础上发展起来的。在该模型中，每个有影响力的节点影响它的邻居的过程是完全独立的，跟其他的节点毫无关系。

下面详细介绍一下消息在该模型上的传播过程。符号说明：S_0 为种子节点集合，S 为有影响力的节点集合，$P(u,v)$ 为节点 u 将节点 v 激活的概率，β 为整个网络的激活阈值。(1) 开始时，$S=S_0$。(2) 接下来，S 中的每个节点 u 以概率 $P(u,v)$ 去影响它的所有邻居(有向图中指出边邻居)，若 $P(u,v)\geq\beta$，则 v 被激活，将 v 加入到 S 中；否则，v 仍保持原状态。当 u 对它的所有邻居 v 都影响了一次之后，便被移出 S，失去了影响能力，既不会影响他人，也不会被他人影响。重复步骤(2)，直到 S 为空停止。

图 2-4 中的网络为 4 个节点的有向图，其中每条有向边上的数值表示节点之

间的影响概率，记为$l(i,j)$。例如节点 2 对节点 3 的影响概率为 0.9，如图 2-4 所示。

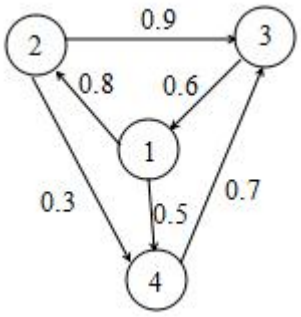


图 2-4 一个 4 个节点的加权有向图

下面，将图 2-4 中所示网络在独立级联模型上进行仿真传播，S_0表示初始的种子节点集，$S_0=\{1\}$，S表示影响力节点集合，激活阈值β=0.8。

初始时$S_0=\{1\}$，节点 1 作为种子节点去影响它的所有邻居节点 2 和 4，因为$l(1,2)=0.8$，所以节点 1 激活了节点 2，将节点 2 加入到 S 中，此时$S=\{1,2\}$；因为$l(1,4)<0.8$，所以节点 1 未激活节点 4，此时节点 1 已将其所有邻居都影响了一次，故失去影响力，移出 S，此时$S=\{2\}$。接着，影响力节点 2 去激活它的所有邻居节点 3 和 4，因为$l(2,3)>0.8$，$l(2,4)<0.8$，所以节点 2 激活节点 3，将节点 3 加入 S，将节点 2 移出 S，此时$S=\{3\}$。接着，节点 3 作为影响力节点去激活它的所有邻居只有节点 1，由于节点 1 是已激活节点，无需再进行激活。此时，将节点 3 移出 S，此时，$S=\phi$，再无影响力节点，传播结束，初始种子节点集$S=\{1\}$共感染了 2 个节点，分别为节点 2 和节点 3。

2.5.3 线性阈值模型

线性阈值模型模型的特点是“共同作用”。某个节点 v 能否被激活，不是取决于它的某一个邻居节点，而是取决于节点 v 的所有处于激活状态的入边邻居节点对节点 v 的共同作用。初始时刻，只有种子集中的节点处于激活状态，其他节点均处于未激活状态。每个节点 v 都有一个激活的临界值θ_v代表该节点被激活的难易度。

S 表示已激活的节点集合，S_0表示种子节点集合。*P(u,v)*为节点 u 将节点 v 激活的概率。初始时刻，$S=S_0$。在信息扩散过程中，对于每一个未激活的节点$v(v\in V\setminus S)$，对于每一个处于未激活状态的节点 $v(v\in V/S)$，考虑它的所有处于激

活状态的邻居节点 u(有向图中为入边邻居)，若 $\sum P(u,v) \geq \theta_v$，则节点 v 被激活，加入到 S 中；否则继续保持原来状态。重复上述过程，直到网络中没有可以被激活的节点。

2.5.4 SIR 模型

SIR 模型是传染病领域的经典模型，起初应用于传染病的抑制。但由于复杂网络中消息的传播类似于传染病的疯狂蔓延过程。因此近年来 SIR 模型被广泛的应用到复杂网络的研究领域中。

在 SIR 模型中，节点共有三种状态，分别是易感状态 S(susceptible),感染状态 I(infected)和免疫状态 R(recovered)。处于易感状态的节点没有患病，因此不能将疾病传染给他人，但有可能被其处于感染状态的邻居传染。处于感染状态的节点已患有疾病，具有感染它的邻居节点的能力，是传染病的传播者。处于免疫状态的节点已经经历过一个完整的感染过程，不会再被感染，也不具有感染其他人的能力。令 β 表示某个易感节点 i 被其处于感染状态的邻居节点 j 感染成功的概率，令 γ 表示感染个体被治愈并获得免疫能力变为免疫个体的概率。SIR 模型节点的状态转换如图 2-5 所示。

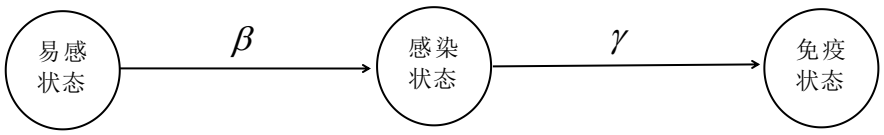


图 2-5 SIR 模型状态转换图

2.6 本章小结

本章主要介绍了复杂网络的相关基础理论。首先介绍了网络及复杂网络的定义。然后介绍了复杂网络中应用最广泛的三个拓扑性质：度分布、平均路径长度和聚类系数。接着介绍了复杂网络领域四种经典的影响力最大化算法并依次分析每种算法的优点和不足。最后，介绍了复杂网络研究领域三种经典的影响力传播模型，详细的介绍了每种传播模型的传播原理和过程，为第 3 章和第 4 章做好理论准备。

第 3 章 基于 k-shell 的影响力最大化算法

在研究网络图的各种性质属性时，研究者往往更多的关注图中度数最大的节点。文献[40]中指出，通过 SIR 模型上的仿真结果，人们发现传统意义上的度最大的节点的传播能力远不及 k-shell 分解算法中最内层的节点。因为即便一个度很大的节点，如果它处在网络的边缘位置，它能影响的节点是非常少的，可能只经过短短几次的口碑相传就停止。因此，k-shell 算法能够有效的识别出网络的核心节点。但是，k-shell 算法只是粗略的将网络划分成了若干层，处于网络核心位置的节点不止一个。因此，如何更细致的衡量 k-shell 层次结构中各个节点的影响力进而合理的找到规模为 k 的种子节点集成为本章研究的重点。

3.1 k-shell 算法介绍及分析

3.1.1 k-shell 算法简介

定义 3.1 k-core。k-core 是指一个所有节点的度都大于或等于 k 的极大子图。

定义 3.2 k-shell。k-shell 指的是一个生成子图，该子图的节点集表示为 $V = V_{(k-1)-core} - V_{k-core}$。

k-shell 是图论里的经典概念。该算法把网络由边缘至中心划分成了若干层，每一层有若干个节点，是一种粗粒度划分网络的方法。其依据的思想就是:处于网络中心位置的节点，即便度很小，由于其位置的优势，影响力也可以很大。而处于边界位置的节点，即便度很大，由于位置受限，其影响力并不大。

k-shell 算法具体分解过程如下：网络中如果存在度为 1 的节点，从度中心性的角度就认为他们是影响力最微弱的节点。把这些度为 1 的节点及其所连接的边都去掉，剩下的网络中就会出现一些新的度为 1 的节点，再将这些度为 1 的节点及其所连接的边去掉。循环操作，直到网络中所有节点的度都大于 1。此时，所有被去掉的节点算作一层，称为 1-壳(记为 k_s=1)。按上述方法继续剥壳，去掉网络中度为 2 的所有节点及其连边，重复这些操作，直到网络中没有度为 2 的节点为止。此时，所有被去掉的节点算作第二层，称为 2-壳(记为 k_s=2)。继续上述操作，直至网络中无剩余节点。在某些网络中可能出现不与任何节点有边连接的节点，即孤立节点。孤立节点的度为 0，故属于 0-壳。网络中的每一个节点属于唯

一的一层，显然在第 k_s 层，所有节点的度都大于 k_s，一个 k-shell 的划分如图 3-1 所示。

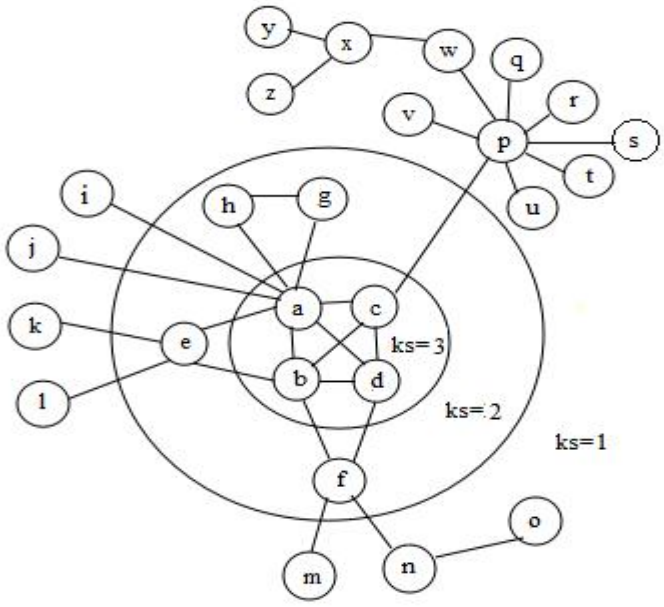


图 3-1 一个可分解为三层壳的简单网络

3.1.2 k-shell 算法问题分析

下面，本文通过图 3-1 中的可分解为三层壳的简单网络分析 k-shell 算法划分中存在的问题进而确定改进的方向。

观察图 3-1 中 k_s=1 这一层，共有 15 个节点，分别是 i、j、k、l、m、n、o、p、q、r、s、t、u、v、w、x、y、z。虽然这些节点的 k_s 值都为 1，但是可以很明显的看出，这些节点的影响力和重要性不尽相同。节点 p 的影响力大于节点 v 的影响力，原因主要有以下两点：(1) 节点 p 的邻居节点更有影响力。节点 p 的一阶邻居节点 c 处于网络核心位置，一旦节点 p 将节点 c 激活成功，便能够获得很大的影响范围。(2) 节点 p 的度更大，节点 p 仅需要一步口碑相传就可以将消息传递给很多人。再看图 3-1 中 k_s=3 这一层，有 4 个节点分别为 a、b、c、d，但是节点 a 的影响力较大，节点 d 的影响力较小，很明显还是因为节点 a 有更多的有影响力的邻居并且度较大。综上分析可知，仅仅以 k-shell 算法中层次划分得到的 k_s 值作为节点影响力的评估指标还不够完美。本文可以在 k-shell 层次结构的基础上，从节点的局部信息考虑，给出节点影响力的更全面的衡量指标。

3.2 基于 k-shell 的影响力最大化 KSLER 算法

3.2.1 单个节点的 KLSC 影响力

在现实生活中，考虑有影响力的人，人们第一想到的就是那些知名度很高的人，比如娱乐圈的电影明星、歌手、体育界的奥运冠军、科学界的著名科学家等等。这些人被很多人崇拜和关注，所以他们的一言一行都有很大的影响力。但是还有一类人，他们本身默默无闻，没有很多人关注他们，但他们周边有很多有影响力的朋友。而以上两种特质都具有的人，也就是不但自身具有影响力，还有很多影响力朋友的人，能够把信息传播的更快更远。

在 k-shell 算法中，节点 v 所处的层数 $k_s(v)$ 表明了节点在整个网络中的重要程度。例如在一个工厂中，员工也被分成了若干层：总经理、部门经理、部门组长和普通员工。显然，总经理的指令是指导整个工厂工作的大方针，所有人都必须遵循。而部门经理的影响范围只是在一个部门内部，而部门组长的影响范围更小，只在他所管辖的组内。而作为普通员工主要做好本职工作就好，其影响范围微乎其微。因此，在网络中，一个节点的位置从根本上决定了该节点的影响力。但是处于同一 k_s 层的节点有很多个，本文还需要考虑更多的因素更细致的对这些节点进行区分。本文考虑节点的最直接的一阶邻居的影响力。同样，节点的邻居节点 u 的影响力主要由其位置 $k_s(u)$ 决定。

已知网络 $G(V,E)$，其中 $V=\{v_1,v_2,...,v_n\}(n=|V|)$ 为节点集，$E=\{e_1,e_2,...,e_n\}(n=|E|)$ 为边集，k-shell 算法将网络 G 划分为 s 层，即 $G=\{K_1,K_2,K_3,...,K_s\}$，并且 $|K_i|\geq 1(i\in N* 且 i\in[1,s])$，记 $k_i(v)(i\in N*,1\leq i\leq s)$ 为节点 v 所在的层次。本章重点就是要对任意的 $v\in K_i$，给出节点 v 影响力 KSLC 的定义式。

节点的核心影响力，是节点影响力的主要构成部分，基于 k-shell 算法划分的准确性，本文将 k-shell 划分的层数作为节点核心影响力的衡量标准。下面，给出节点核心影响力的定义。

定义 3.3 节点核心影响力。节点的核心影响力即节点的主要影响力。本文将任意节点 v 的核心影响力定义为 k-shell 算法将网络划分后节点所在的层数，即 $k_S(v)\,(1\leq k_s(v)\leq s)$。

以上本文定义了节点的核心影响力，根据问题分析中提出的问题。本章还需

要考虑节点的局部信息，下面给出了节点局部影响力的定义。

定义 3.4 节点局部影响力。节点 v 的局部影响力即节点的次要影响力，指的是节点 v 直接相连的一阶邻居的影响力，包括其一阶邻居 u 的位置(即 $k_s(u)$)和度(在有向图中指出度)，记为 $neighbor(v)$，定义见式(3-1)。

$$neighbor(v) = \sum(k_s(u) + bt(u)) \tag{3-1}$$

式中 $1 \le k_S(u) \le s$， $0 \le bt(u) \le 1$。

为了表示节点的 k_s 值对影响力的主要贡献，本文将节点的度的范围控制在 0~1 内，其统一化的定义见式(3-2)。

$$bt(v) = \frac{\deg(v)}{\max(\deg(u))}(u, v \in V) \tag{3-2}$$

定义 3.5 节点全局影响力。本文定义节点 v 的全局影响力分为两部分，一是节点 v 的核心影响力，即 $k_S(v)$；二是节点 v 的局部影响力。用 KSLC(v)表示节点的全局影响力，定义见式(3-3)。

$$KSLC(v) = k_s(v) * neighbor(v) \tag{3-3}$$

运用公式(3-1)展开，结果见式(3-4)。

$$KSLC(v) = k_s(v) * \sum(k_s(u) + bt(u)) \tag{3-4}$$

根据以上的定义式，本章给出了单个节点影响力的度量指标。但根据影响力最大化问题的定义，需要找到的是一个规模为 k 的种子节点集合，这个集合的能够影响的节点数最多。这里就有一个问题，将若干个单个最优的节点放在一些，能达到整体最优吗，下面本文借助图 3-1 小型的实例网络分析一下。

按照上述公式(3-4)中节点全局影响力 KSLC 的定义，计算该网络最有影响力的 top5 依次是：a、b、c、d、e。但是本文发现，a、b、c、d 互为邻居节点，在信息扩散过程中，节点 a 非常有可能将 b、c、d 中的若干个激活成功，同样 b、c、d 也是。针对这个网络而言，选择 a 就可以达到 a、b、c、d 这四个节点的影响范围，所以 b、c、d 的选择有些浪费。那么这样就造成了种子节点的聚堆现象，导致种子节点集的影响力因重叠而变弱。

从上面的实例中本文可以发现，单个最优节点的并集存在一个弊端，那就是影响力的节点会出现“聚堆”现象，尤其对于无向图来讲，这种弊端暴露的更明显。k-shell 划分后同处于网络核心的若干节点之间互为邻居，互相贡献影响力，导致这些节点的 KSLC 值都比较高而被选中做了种子节点。这样就造成了影响力

的重叠和种子节点的浪费。因此，本文需要从集合的整体来考虑，如何由单个最优得到整体最优。

3.2.2 种子节点集的整体优化策略

(1) 优化分析

在影响力最大化问题中，通常都是根据节点的某个值或者定义式(可以叫做能量)来选择该值最高的 k 个节点作为种子节点。但是本文发现，当某个节点 v 被选做种子节点后，与节点 v 直接相连的一阶邻居很容易被激活成功，因而也导致节点 v 的一阶邻居的邻居即节点 v 二阶邻居被激活成功的概率也大于其他节点。因此在选择种子节点时，一旦某个节点 v 被选中，则应尽量减少再选择 v 两步之内的节点作为种子节点。但是如果把 v 周围的节点全部覆盖掉的话，一方面可能会导致某些能量非常大的节点丢失；另一方面，对于一个密集网络，可能会覆盖掉很多的节点而导致种子节点的数目达不到 k 值的要求。

(2) 优化思想

针对上述分析，本文提出一种对于种子节点集的局部能量缩减的优化思想，给出节点的缩减度 *reduction*，根据节点与种子节点的距离做不同程度的缩减。另外，在有向网络中，节点的入度表示该节点在信息传播过程中能被其他节点激活的次数，显然，被激活的次数越多被激活成功的概率越高，所以有向网络中那些入度较小的节点很难被激活成功。因此，在选择种子节点时，应优先选择那些传播能力强但被激活能力弱的节点。

3.2.3 算法详细描述

综合定义 3.4 中给出的单个节点影响力 KSLC 的定义及 3.2.2 中对种子节点集的优化策略，本文提出一种基于 k-shell 局部结构的能量缩减算法 KLSER(Kshell-based Local Structure Energy Reducted Algorithm)。

算法主要分为两大步：

(1) 对网络中的每个节点计算其影响力 KSLC 值，并进行排序。

(2) 每次选择 KSLC 值最大的节点中入度最小的节点加入到种子集合并将该节点的能量至 0，然后对该节点的一级邻居和二级邻居进行能量缩减。重复此步骤，直到选择的种子节点的个数为 k。

在算法的具体实现方面，本文在 Matlab 2015b 的软件环境下编写代码，用邻接矩阵存储网络图。在种子节点集整体优化时，规定被选做种子的节点的一阶邻居缩减度为 $reduction$，二阶邻居的缩减度为 $1-reduction^2$。另外对于无向图，入度和出度是一个概念，即节点的度。故在考虑惰性节点时，以节点的度作为衡量指标。

下面给出 KLSER 算法的伪代码。基于 k-shell 局部结构的能量缩减算法 KLSER(Kshell-based Local Structure Energy Reducted Algorithm)描述见算法 3.1。

算法 3.1 KLSER 算法

输入：节点的入度 in_degree，节点的 KSLC 值，缩减百分比 *reduction*(0 到 1 之间)
　　种子节点的个数 k

输出：种子节点集合 seed

```
BEGIN
(1)  count=0;
(2)  for v∈V do
(3)        KLSER(v)=KSLC(v)+1/(indegree(v)+1);
(4)  end for
(5)  while count< k
(6)        v1=max(KLSER);                    //找出 KLSER 最大的节点 v1
(7)        seed <- v1;                       //将 v1 加入种子节点集
(8)        KLSER(v1)=0,                      //v1 能量置 0
(9)        count=count+1;                    //种子节点个数加 1
(10)       for u∈N(v1) do                    //对 v1 的一阶邻居能量缩减
(11)            KLSER(u)=reduction*KLSER(u);
(12)            for w∈N(u) do                //对 v1 的二阶邻居能量缩减
(13)                 KLSER(w)=(1-reduction)^2*KLSER(w);
(14)            end for
(15)       end for
(16)end while
END
```

算法 3.1 通过综合考虑了节点的 k_s 值和节点的邻居信息给出了网络中单个节

点影响力的衡量指标，即节点的 KSLC。在考虑节点邻居信息时，不仅考虑了邻居节点的位置(即 k_s 值)还考虑了邻居节点的出度，有效的解决了 k-shell 算法划分粗糙的问题。另外，对比其他基于 k-shell 的改进算法，本算法从种子节点集的整体角度考虑，选择传播力强又不易被感染的节点作为种子节点，并采用能量缩减策略有效解决了其他算法中存在的种子节点影响力的重叠问题。从算法效率的角度看，对于一个具有 n 个节点 m 条边的复杂网络，算法整体的复杂度为 $O(n^2)$，能够适用于当前的大规模网络。

3.3 KLSER 算法实例分析

针对图 3-1 的例子，本章提出的 KLSER 算法得到的规模为 5 的种子节点集为 S={a、d、e、x、f}，可见选出来的种子节点分布的比较均匀，没有出现聚堆现象。

对图 3-1 中的人工网络分别按节点的度、节点的 k_s 值、公式(3-4)中定义的节点的全局影响力 KSLC 作为衡量指标，选择值最高的 5 个节点作为种子节点加入到种子集合中与本章提出的 KSLER 算法得到的 k=5 的种子节点集进行对比。如表 3-2 所示。

表 3-2 不同识别算法选出的 k=5 的种子节点集

	Degree	k-shell	KSLC	KLSER
1	*a*	*a*	*a*	*a*
2	*p*	*b*	*b*	*d*
3	*b*	*c*	*d*	*e*
4	*c*	*d*	*c*	*x*
5	*d*	*e*	*e*	*f*

从表 3-2 中可以看出，度中心性算法仅以节点的度作为衡量指标，认为节点 p 的影响力仅次于节点 a，显然不够客观。因为如果节点 p 不能激活节点 c，那它的影响力只能局限在较小的范围，最多能感染 q、r、s、t、u、v、w、x、y、z 这 10 个节点。k-shell 和 KSLC 算法选择的种子节点集几乎一致，只是在节点 c 和节点 d 的影响力上有些差异，综合来看，节点 d 的影响力优于节点 c，KSLC 指标考虑的更全面。但这两个算法都选择了 a、b、c、d 这四个节点，造成了影响力的严重重叠现象。KLSER 算法既选择了影响力较大的 a、d 节点，又将有影响力的节点分散开来，保证了每个节点的影响范围，使种子集整体发挥出最大的影响范围。

3.4 本章小结

本章首先详细介绍了 k-shell 算法的步骤及原理，并借助具体的人工网络分析了 k-shell 算法的不足之处，即划分粗糙，节点影响力定义模糊。接着，本章针对 k-shell 算法的问题并结合真实网络的信息传播机制，提出了一种基于 k-shell 的局部能量缩减算法。给出了相关的定义详细的算法描述和伪代码，并对算法的效率和复杂度进行了简单的分析。最后，在一个小型人工网络上进行实例分析，对比度中心性算法、k-shell 算法和 KLSER 算法选出的 top5 种子节点集，并从网络结构上进行分析。

第 4 章 基于社区结构的影响力最大化算法

“社区”并没有一种严格的定义，本文可以将其理解为一类具有相同属性的节点的集合。社区内节点间连接密切，社区与社区之间连接稀少。因此在社会网络中，可以根据不同社区的人群特点，制定个性化的营销策略。除此之外，消息在社区内会传播的更为迅速。因此，可以借助网络的社区结构进行信息的快速病毒式传播。另外，面对网络规模的快速增长，网络结构的愈加复杂，将复杂网络划分成若干社区可以降低处理问题的空间复杂性和时间复杂度。综上所述，研究社区结构具有重要的理论意义和实践价值。另外，具有 $O(n^2)$ 复杂度的 k-shell 算法在小型的社区结构上运行的更快，能够更精确的找出网络的核心。因此，本章结合 k-shell 算法，研究社区结构上的影响力最大化算法。

4.1 社区结构特性

社区是现实生活的真实反映。当人们通过网络聚集在一起时，彼此之间互相联系，当一部分节点因为某种共同的属性而紧密联系到一起时，就构成了一个社区。社区结构是复杂网络的一个至关重要的特性。社区是指网络中的一个特殊的节点集合，该集合内的节点之间具有较多的共同点和相似性并且节点之间的连边较多。从图论的角度来看，社区就是网络图中的一个子图，子图内的节点连接紧密，但与子图外的节点连接疏松。一般在对网络进行社区划分时，由于网络结构的多样性，社区的大小是不确定的。

真实复杂网络中比较典型的社区结构的例子就是空手道俱乐部中高层领导之间的决裂形成的社区结构。空手道俱乐部网络是 Zachary 通过观察真实的美国某大学空手道俱乐部 34 个成员及其之间的关系而建立的。节点代表俱乐部的每个成员，边表示成员之间的朋友关系，并且只有在俱乐部内部和外部交往都很频繁的朋友之间才会有连边。由于校长和主管之间的意见分歧，导致网络中产生了两个意见不同的拥护团体。如图 4-1 所示。节点 1 表示俱乐部的主管，节点 34 表示俱乐部的校长。白色的节点表示拥护校长的俱乐部成员，深灰色的节点表示拥护主管的俱乐部成员。观察图 4-1 可以发现，两个团体内部成员间联系密切，连边错综复杂。而两团体之间只有 8 条连边，数目相对较少。

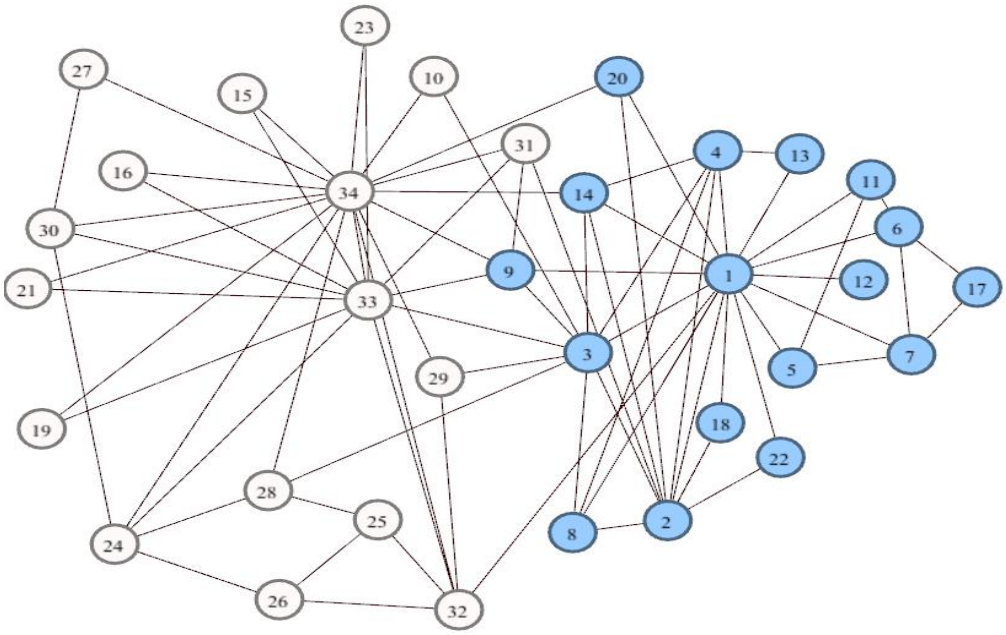


图 4-1 空手道俱乐部社区结构图

4.2 Louvain Method 社区划分算法

Louvain Method 算法是一种基于贪心思想的社区发现算法，该算法尝试将每个节点放入其邻居节点所在的社区并通过衡量模块度的增量变化来确定是否合并该节点及合并到哪个社区。该算法已经在很多不同类型的网络中得到了成功的应用，在标准 PC 机上分析出一个 200 万节点的网络仅需要两分钟。该算法是当前大型网络中社区划分领域应用最广泛的一种方法。

算法步骤：

(1) 令每个节点为一个单独的社区，初始时社区数目等于网络节点总数；

(2) 将每个节点 i 放入其邻居所在社区并计算加入前后的模块度变化量 ΔQ，若 ΔQ 大于 0，则将节点 i 加入到 ΔQ 最大的社区。否则，节点 i 保持原社区不变。

(3) 重复 2)，直到所有节点的所属社区不再变化；

(4) 对网络图进行简化，将每个社区的所有节点抽象成一个节点，看社区之间是否还有合并的可能。如果有，对抽象之后的图，继续进行 2)和 3)，直到整个网络的模块度不再变化。整个网络的模块度变化如式(4-1)所示。

$$\Delta Q = \left[\frac{\sum in + k_{i,in}}{2m} - \left(\frac{\sum tot + k_i}{2m} \right)^2 \right] - \left[\frac{\sum in}{2m} - \left(\frac{\sum tot}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \tag{4-1}$$

Louvain Method 算法具有以下优势。

(1) 算法得到的社区结构是分层的，每一轮计算完成后得到的新图都是对一个大社区内若干细分社区发现的结果，这样的分层结构是每个网络的自然属性，使研究人员得以深入了解某个社区内部结构和形成机制。

(2) 该算法易于实现，并且计算过程全程无监督，即最终结果完全依赖于算法聚类，并不需要人为提前预设分类。

(3) 算法的性能较好，在进行一些经典社区分类算法的对比中，Louvain Method 算法对图的大小几乎没有上限要求，并且能在迭代几轮后快速收敛。这为处理拥有百万级别以上节点的移动通信网络甚至上亿节点的大型社交网络的社区发现提供了可能。

这里，先利用 Pajek 软件中集成的 Louvain Method 社区划分算法将大型复杂网络 G 划分为若干个社区，然后在社区结构的基础上进行影响力最大化算法的研究。

4.3 基于社区结构的影响力最大化 IBC 算法

4.3.1 问题的提出

社区结构是社交网络的一个普遍特性。在社区内部，节点之间交流比较密切，但是在社区之间，节点之间交流相对较少。在处理实际问题时，常常会遇到的网络规模异常庞大网络结构非常复杂，用一般的影响力最大化算法处理起来复杂度太高，耗时较多。社区结构这一特性给了本文很好的启发，本文可以对复杂网络进行分块处理，把大型复杂网络分解成若干个部分，在这里就是若干个社区。先在每个社区内部研究影响力最大化问题，然后把整个网络的所有社区综合并连接起来，研究整个网络上的影响力最大化问题。

实验表明，k-shell 算法能够快速并且有效的找出网络的核心，尤其对于规模较小的网络尤为适用。除此之外，减小网络的规模有助于降低 k-shell 层次划分的粗糙度，使 k-shell 算法对节点影响力的评估更准确。因此，本文可以对网络中的每个社区进行 k-shell 分层，将网络中所有节点按 k_s 值降序排列，找出每个社区内部的核心节点集。此外，还需将消息在社区之间快速传播。因此还需要一些重要边界节点。这样，信息就既可以在社区内快速蔓延，又可以在社区之间广泛传播。

4.3.2 问题描述

已知网络 $G(V,E)$，其中 $V=\{v_1,v_2,...,v_n\}(n=|V|)$ 为节点集，$E=\{e_1,e_2,...,e_n\}(n=|E|)$ 为边集，Louvain Method 算法将网络划分为 M 个社区，分别记为 $C_1,C_2,...,C_M$，即 $G=\{C_1,C_2,...,C_M\}$。k-shell 算法将每个社区 $C_i(1\le i\le M)$ 划分成了 k_i 层，对任意的节点 $v\in C_i$，本文用 $k_s(v)$ 表示社区 C_i 用 k-shell 算法划分节点 v 所在的层数。本文的目的就是找到每个社区 i 的核心节点集 $F_{C_i}(F_{C_i}\subset C_i)$ 和整个网络的边界节点集 B_G，从 $\bigcup^{M}F_{c_i}$ 和 B_G 中选择若干节点构成种子节点集 S，使得 S 中的节点能够感染的节点数最多。

4.3.3 算法思想及相关定义

什么样的节点才可以算是社区间的重要枢纽节点呢？又该怎样衡量节点的在社区间传播信息的能力呢？简单来看，本文一般认为，节点 v 到很多个社区都有连边，那么节点 v 就有机会将信息传播到这些社区，所以连接社区的数目可以作为一项衡量边界节点影响力的一项指标。另外，社区间的每条连边的重要性也不同。例如社区 C_1、C_2，u_1 是社区 C_1 内的点，v_1、v_2 是社区 C_2 内的点，且 $k_s(v_1)$=2，$k_s(v_2)$=8。存在社区 C_1、C_2 之间连边 (u_1,v_1) 和 (u_1,v_2)，很明显，边 (u_1,v_2) 更重要，因为 v_2 较 v_1 更接近于 C_2 的核心，边 (u_1,v_2) 与 C_2 的核心相连接，会激活更多的节点 。另外，连边的重要性还与其所连接的社区的规模大小有关，连接的社区规模越大，该连边能够激活的节点越多，该连边越重要。以上所述的都可以作为评价边界节点枢纽重要性的指标。

(1) 边界节点枢纽度

网络建模：网络用图 $G(V,E)$ 表示，其中 $V=\{v_1,v_2,...,v_n\}(n=|V|)$ 为节点集，$E=\{e_1,e_2,...,e_n\}(n=|E|)$ 为边集，设网络中社区个数为 M，则所有社区集合为 $C=(C_1,C_2,...,C_M)$。

定义 4.1 社区内部节点。在社区 C_i 中，以 $v(v\in C_i)$ 为端点的边的另一端点 u 也在社区 C_i 中，则这样的节点 v 为社区内部节点。

定义 4.2 边界节点。在有向图中，对于任意节点 v，若存在有向边 (v,u) 其中 $v\in C_i$ 并且 $u\in C_j\ (i\ne j)$，则称节点 v 为边界节点。在无向图中，u 和 v 都是边界节点。简单来说，边界节点指那些始点和终点不在一个社区内的节点。

从定义 4.1 和定义 4.2 可以看出，在包含若干社区的网络 G 中，只有边界节点具有连接多个社区的能力，能将信息从一个社区传播到另一个社区。而社区内部的节点只能将信息在本社区的内部进行传播。本文用 k-shell 算法来评价社区内部节点的重要性，那么如何去评价边界节点的重要性呢？边界节点的特殊性就在于其连边跨越了两个不同的社区。所以评价边界节点主要是评价其在社区间连边的重要性。

对于大规模的复杂网络，每个边界节点 i 可能会拥有很多跨越社区的连边。对于有向图，这些连边的起点为 $i(i \in C_i)$，而终点 j_k 属于不同的社区 $C_k(k \neq i)$。当连边的终点 j_k 在其所在的社区 C_k 中处于一个不利于传播的位置时(例如 $k_s(j_k)$ 很小)，那么即便该连边 (i, j_k) 连接了社区 C_i 和 C_{j_k}，但不能有效的将消息从社区 C_i 广泛的传播到社区 C_{j_k}。因此，社区间的有向连边的终点很重要，连边的终点也必须是一个有影响力的节点，这样边界节点 i 的连边才有传播能力。也就是说连接到社区核心的边要比连接到社区边界的边更重要。下面针对于此，本文给出有效节点和有效连接的定义。

定义 4.3 有效节点。已知网络 $G=(V,E)$，社区 C_i，k-shell 算法将 C_i 划分为 k_i 层，若任意节点 $v \in C_i$ 且 $k_s(v) \geq \frac{k_i}{2}$,则称节点 v 为社区 C_i 的有效节点，记 S_{C_i} 为社区 C_i 的有效节点集合。

定义 4.4 有效连接。已知网络 $G(V,E)$，社区 $C_{i,}C_j(i \neq j)$，若存在边 $(u,v)(u \in C_{i,}v \in C_j)$ 且 $v \in S_{C_i}$，则称边 (u,v) 为有效连接。简单来说，即边的终点为有效节点的连接为有效连接。

通过定义 4.3 和定义 4.4 就可以去除掉一些社区间没用的连边，简化社区间的连接情况。以后，本文只考虑社区间的有效连接。

对于边界节点 i 在不同社区 C_{j_k} 间的有效连接，它们的重要性也不同。对于从起始点 i 连接到不同社区 C_{j_k} 的连边。本文认为，两个社区间连边越多，则这两个社区之间的边权值越大，因为一条路径的联系可以带动出更多条路径的联系，从而使两社区沟通更紧密。其次，每条连边的权重还与该边连接的社区的规模(节点数目)有关，连接的社区规模越大，这条连边能够激活的节点越多，因此越重要。本文根据连边所连接的两社区 C_i 和 C_{j_k} 的联系的紧密程度，即两社区的连边数目，给出社区联系度的定义，并进一步定义社区间连边的权值。

定义 4.5 社区联系度。表示任意两个社区联系的紧密程度。用 $H(i,j)(i,j\in[1,M])$ 表示，具体定义见式(4-2)。

$$H(i,j)=\frac{T(i,j)}{|C_i|} \tag{4-2}$$

其中 $T(i,j)$ 表示社区 C_i 到社区 C_j 的有效连接数。

定义 4.6 社区间连边的权值。对于任意的社区间的连边 (u,v) 其中 $u\in C_i$ 并且 $v\in C_j$，定义连边 (u,v) 的权重不仅与社区 C_i、C_j 的社区联系度 $H(i,j)$ 有关，还与终点 v 所在的社区的规模 $|C_j|$ 有关。首先给出社区规模 num_j 的统一化定义标准，见式(4-3)。

$$num_j=\frac{|C_j|}{\sum_{i=1}^{M}|C_i|} \tag{4-3}$$

下面，本文用 $w(u,v)$ 表示连边 (u,v) 的权重，具体定义见式(4-4)。

$$w(u,v)=H(i,j)+num_j \tag{4-4}$$

以上，本文分析了影响社区间连边的重要性的因素并定义了社区间连边的权重，接下来，本文给出边界节点枢纽重要度的定义。

定义 4.7 边界节点枢纽度。表示边界节点 u 连通其他所有社区能力的强弱。用 Link(u)表示，具体定义见式(4-5)。

$$Link(u)=\sum^{M}\sum w(u,v) \tag{4-5}$$

到目前为止，本文已经给出社区内部节点和整个网络边界节点的重要性衡量指标。接下来本文要研究的就是对于规模为 k 的种子节点集，需要从所有社区内部的节点集选取节点数目的比例 t(由第 5 章实验获得)和具体到每个社区 C_i 需要选取的节点数目。

众所周知，各社区内的核心节点是整个社区的核心，对该社区内的信息传递起到了决定性的作用。因此选取的比例 t 应较大一些。而重要的边界节点将四分五裂的网络联系起来，也起到了不可替代的作用，所以必须选择重要的边界节点作为种子节点，但是数量上应该做到少而精。

(2) 节点选取比例的确定

在基于社区结构的影响力最大化算法的研究中，很多学者在确定从各个社区选取的种子节点数目时，都是以社区的规模作为衡量指标。首先，这个衡量指标

有一定的合理性。规模较大的社区，显然需要较多的种子节点去感染，这样才能达到比较好的影响范围。而规模较小的社区，仅仅需要少数的几个种子节点去感染就能够获得不错的感染范围。但是这种思想有一定的局限性，该思想仅仅考虑了社区内部的核心节点对整个社区的感染能力，而忽略了网络的边界节点对各个社区的激活能力。另外，很多社区划分算法划分的社区规模比较均匀。显然，这时以社区规模来确定种子节点的数目更显得苍白无力。基于上述分析，本文提出以社区被其他社区激活的程度来确定该社区选取种子节点的个数。

定义 4.8 社区激活度。表示社区 C_i 被所有其他社区激活的程度。用 ACT_{C_i} 表示，具体定义见式(4-6)。

$$ACT_{ci} = \frac{\sum_{j\neq i, j=1}^{M} L(j,i)}{|C_i|} \tag{4-6}$$

其中，$L(j,i)$表示 j 社区到 i 社区的有效连接数目，$|C_i|$表示社区 C_i 的节点个数。

从定义 4.8 可以看出，社区激活度表示社区被其他所有社区的激活能力。显然，容易被其他社区激活的社区当然可以少选择一些社区内部核心节点作为种子节点。

定义 4.9 种子节点集 Seed。网络 $G=(C_{1,}C_{2,}C_{3,...,}C_M)$,每个社区内的核心节点集为 $F_{C_i}(i\in[1,M])$，网络 G 的边界节点集为 B_G,则网络 G 的种子节点集 *Seed* 包含这两部分，具体定义见式(4-7)。

$$Seed = \bigcup_{i=1}^{M} F_{C_i} \cup B_G \tag{4-7}$$

设从核心节点集 $\bigcup^{M} F_{C_i}$ 中取节点数的百分比为 $t\in(0,1)$，则从边界节点集 B_G 中选取节点的比例为 1-t，种子节点集的组成见式(4-8)。

$$|seed| = t\,|\bigcup_{i=1}^{M} F_{C_i}| + (1-t)\,|B_G| \tag{4-8}$$

接着，根据社区激活度 ACT_{C_i} 来确定每个社区需要选取的核心节点数， 社区激活度越高的社区，该社区被其他社区激活的节点数越多，因此社区内核心节点集可以少取一些。各个社区选取的种子节点数满足比例见式(4-9)、(4-10)和(4-11)。

$$|F_{C_1}|:|F_{C_2}|:...:|F_{C_M}| = BCT_{C_1}:BCT_{C_2}:...:BCT_{C_M} \tag{4-9}$$

$$BCT_{C_i} = 1 - ACT_{Ci}(i \in [1, M]) \tag{4-10}$$

$$A_i = \frac{BCT_{Ci}}{\sum_{j=1}^{M} BCT_{C_j}} \tag{4-11}$$

综合公式(4-9)、公式(4-10)和公式(4-11)，得到公式(4-12)。

$$|seed| = t\sum^{M} A_i |F_{C_i}| + (1-t)|B_G| \tag{4-12}$$

4.3.4 算法详细描述

在 4.3.2 和 4.3.3 已给出衡量社区内节点和社区边界节点重要性的方法和种子节点选取比例的依据。本文提出一种基于社区结构的影响力最大化 IBC 算法(Influence maximization algorithm based on the structure of the community)。该算法主要分为两大步骤：

(1) 在每个社区上进行 k-shell 层次划分，并对节点按 k-shell 值降序排列。

(2) 根据边界节点枢纽度的定义，计算网络中每个节点的边界枢纽度 link(v)，并按 link(v)值降序排列。

(3) 根据公式(4-6)计算社区激活度，并根据公式(4-9)、(4-10)、(4-11)和(4-12)算出每个社区和边界节点集选择的节点数目(种子节点集的规模为 k)。

(4) 根据(3)中的比例分别从(1)和(2)中降序排列的节点集中选择节点加入到种子节点集合。

在算法的具体实现方面，本文在 Matlab 2015b 的软件环境下编写代码，用邻接矩阵存储网络图。本算法对无向图也适用，在具体计算时，认为边 (u,v) 和边 (v,u) 是两条不相同的边。还有就是在计算选取节点的比例时，经常会出现小数，本算法采用的是四舍五入的方法，并且先选择社区内节点，剩下的节点从边界节点集中取。下面，本文给出 IBC 算法的伪代码表示。

算法 4.1 IBC 算法伪代码

输入：网络 $G=(V,E)$，社区结构 $G=(C_1,C_2,C_3,\ldots,C_M)$，比例 t，种子节点集规模 k。

输出：种子节点集 Seed

BEGIN

(1) for(i=1 to M do)

(2) ks(i)=k-shell(Ci); //对每个社区进行 k-shell 分层

(3) end for
(4) for(i=1 to M do) //计算任意两社区的联系度
(5) for v∈C_M
(6) t=Findneighbor(v); //找到 v 的所有有效邻居节点
(7) j=findcommunities(t); //判断 v 的邻居节点所属的社区号 j
(8) if (i≠j) //若两节点不属于同一社区
(9) count(i,j)++; //i 社区和 j 社区之间的连边数加 1
(10) end if
(11) end for
(12) end for
(13) for u=1 to n do //计算社区间连边的权值
(14) for v=1 to n do
(15) i=findcommunities(u); //判断节点 u 所在的社区
(16) j=findcommunities(v); //判断节点 v 所在的社区
(17) if i≠j and (u,v)=1 //若两节点在不同的社区并且之间有边
(18) w(u,v)=count(i,j)/numi+numi/sum(num); //根据 4-3 计算连边权值
(19) end if
(20) end for
(21) end for
(22) for(u=1 to n do) //计算每个边界节点的枢纽重要度
(23) for i=1 to M do
(24) for v≠C_i
(25) if (u,v)=1
(26) Link(u)=link(u)+w(u,v);
(27) end if
(28) end for
(29) end for
(30) end for
(31) for(i=1 to M do) //计算每个社区的社区激活度
(32) $ACT(i) = sum(count(:,i))/|C_i|$

(33) end for

(34) 根据公式(4-9)、公式(4-10)、公式(4-11)和公式(4-12)从各个社区的核心节点集和边界节点集中选取若干节点作为种子节点加入到种子节点集。

END

算法综合考虑了社区结构中社区内节点和社区间边界节点对影响力传播的共同作用。尤其在边界节点影响力定义中，考虑边界节点到各社区连边的潜在激活能力定义了连边的权值，并根据此给出了边界节点影响力定义式。在确定各社区选取种子节点比例时，改进了其他算法只考虑社区规模的做法，以该社区被边界节点激活的范围作为选取比例的依据，更贴近实际的传播过程。从算法效率看，在计算社区联系度时循环次数为 $M*|C_i|\approx n$，复杂度为 $O(n)$，计算边界节点枢纽度，涉及到三层循环的嵌套，复杂度为 $n*M*|C_i|$，其中 $M*|C_i|\approx n$，因此整个算法的时间复杂度为 $O(n^2)$，能够适用于大规模网络。

4.4 实例分析

本文在一个小型的人工网络上分析基于社区结构的 IBC 算法具体实现。该网络共有 20 个节点，41 条边，是一个无向无权网络。该网络共有 3 个社区，各社区包含的节点数分别为 7、8、5。如图 4-2 所示。

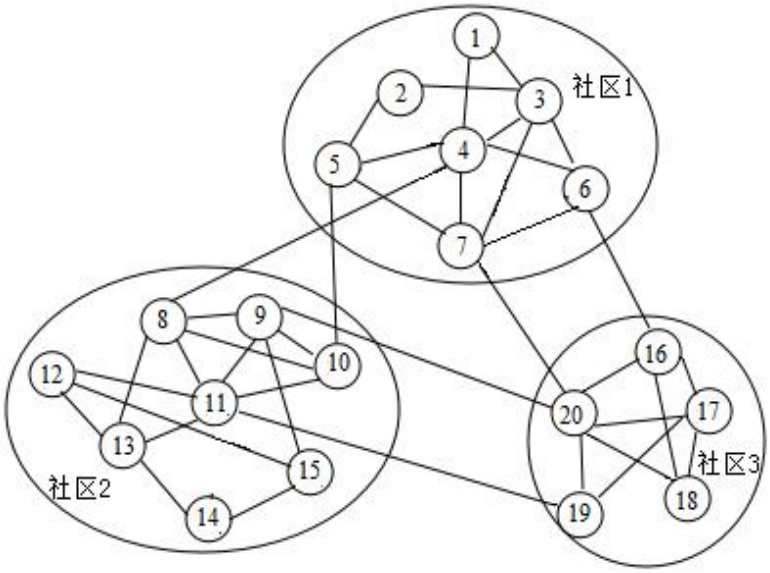


图 4-2 一个简单的包括 3 个社区的人工网络

接着，对每个社区进行 k-shell 划分，每个社区被划分的情况如表 4-2 所示。

表 4-2 3 个社区 k-shell 划分情况

社区编号	节点数	k-shell 层数	第一层节点	第二层节点
1	7	2	1、2、5	3、4、6、7
2	8	2	14、15、12、13	8、9、10、11
3	5	2	19	16、20、17、18

由于网络规模较小，导致 k-shell 划分的层数较小，因此每一条社区间的连边(4，8)、(5，10)、(6，16)、(7，20)、(9，20)、(11，19)都是有效连接。

依照公式(4-2)，计算得到任意两社区的社区联系度如表 4-3 所示。

表 4-3 各社区间的联系度计算结果

	1	2	3
1	0	2/7	2/7
2	2/8	0	2/8
3	2/5	2/5	0

根据公式(4-4)计算得到各边的权值如表 4-4 所示。

表 4-4 社区间连边的权值计算结果

边	权值	边	权值
(4,8)	0.69	(8,4)	0.6
(5,10)	0.69	(10,5)	0.6
(6,16)	0.54	(16,6)	0.75
(7,20)	0.54	(20,7)	0.75
(9,20)	0.5	(20,9)	0.5
(11,19)	0.5	(19,11)	0.8

结合表 4-4 中的计算结果，根据公式(4-5)计算得到边界节点 4、5、6、7、8、9、10、11、16、19、20 的枢纽重要度分别为 0.69、0.69、0.54、0.54、0.6、0.5、0.6、0.5、0.75、0.8、1.25。根据公式(4-6)计算各个社区的激活度，并根据公式(4-9)、公式(4-10)、公式(4-11)和公式(4-12)计算得到社区 1、2、3 都分别从各自社区内选择 1、2、1 个社区内核心节点作为种子节点，分别为节点 3、8、9、16。然后根据计算从边界节点集中选择一个重要边界节点，为节点 20 加入到种子集合，因此最后的种子节点集为 $S=\{3,8,9,16,20\}$。

4.5 本章小结

本章首先详细的介绍了网络的社区结构，并举出了真实的空手道俱乐部网络分裂成社区的实例。接着详细的介绍了 Louvain Method 社区划分算法的原理步骤及其优势。然后，结合 Louvain Method 社区划分算法，提出一种基于社区结构的影响力最大化 IBC 算法。从整个网络的社区结构和信息真实的传播角度给出了相关定义，并给出算法的详细描述和伪代码表示。然后分析了整个算法的效率及复杂度，并在人工网络实例上给出整个算法的简单实现和计算流程。

第 5 章 实验结果及分析

本章将对第3章和第4章提出的两种算法进行仿真对比实验以验证其有效性。

5.1 实验环境

(1) 硬件环境

PC 机，win7 操作系统，Intel(R)Core(TM) i5-4590 CPU 处理器，安装内存 4.00GB，64 位操作系统。

(2) 软件环境

Matlab 2015b 和 Pajek。

Matlab 是一款常用的数学编程软件。和以往的编程软件所不同的是，Matlab 的基本数据单元是矩阵，所有的数据运算都要转化成矩阵的运算来完成。因此 Matlab 可以用来解决复杂的数据运算，并且要比 C、C++等编程语言简洁的多。例如 C 语言里的一个复杂的二层循环在 Matlab 中就是一个简单的矩阵乘法。本文的算法实现和实验结果图的绘制均采用 Matlab 软件。

Pajek 是分析大型复杂网络的有效工具，主要集成了一些复杂网络基本的统计量的计算算法。本文采用了 Pajek 软件中集成的 Louvain Method 算法进行社区的划分。

5.2 仿真模型及评价标准

本章中实验采用了独立级联模型(Independent cascade model)作为传播模型。为了保证实验结果的准确性，对每个数据集都进行 100 次仿真实验，然后取平均值作为实验结果。每次实验都会随机产生一个 $N*N$ 的激活概率矩阵 P，$P(u,v)$表示节点 u 激活节点 v 的概率。在模型的实现上，采用队列结构表示有影响力的节点集合 S，每当有新节点 i 被成功激活时，将节点 i 加入到队尾。当某个节点 j 已经激活完它的所有邻居节点时(不考虑是否激活成功)，将节点 j 移出队列。当队列为空时，也就是网络中再无具有影响力的节点，此时仿真结束。

本章中所有实验都以种子节点集合在仿真结束后总共激活的节点数目即影响范围作为评价标准。激活节点数目多的算法性能高，选出的种子节点集的质量好。

5.3 KLSER 算法的实验结果及分析

5.3.1 实验数据集

Jazz musicians 爵士音乐家合作网络[41]。该网络共有 198 个节点和 2742 条边，其中每个节点代表着一位音乐家，每一条边代表着两位音乐家曾经合作过，是一个无向无权网络。

Neural network Dataset 神经元网络，该网络表示一种名叫 Caenorhabditis elegans 的线虫所具有的有向神经元，由 D.Watts 和 S.Strogatz 作为网络数据集引入。该网络共有 297 个节点，2148 条边，是一个有向无权网络。两个网络的基本信息见表 5-1。

表 5-1 两个真实网络的统计信息

序号	数据集	特点	节点数	边数	平均度
1	Jazz musicians	无向无权图	198	2 742	13.85
2	Neural network	有向无权图	297	2 148	7.93

实验选取了 Neural network 和 Jazz musicians 这两个真实网络数据集作为测试集，用度中心性算法、紧密中心性算法、介数中心性算法和 k-shell 算法作为对比算法，以种子节点集最终感染的节点数目(影响范围)做为评价指标，探究不同算法选取的种子节点集的质量。

5.3.2 实验结果分析

第 3 章提出了基于能量缩减思想的种子节点集优化策略，即当某个节点被选做种子节点时，该节点的一级邻居和二级邻居的 KSLC 值需要做不同程度的缩减，缩减度记为 reduction $(0 \le reduction \le 1)$，reduction 的取值需要从实验获得。Jazz musicians 网络和 Neural network 网络探究 KLSER 算法中 reduction 取值对感染范围的影响(这里激活阈值 $\beta = 0.9$)如图 5-1 和图 5-2 所示。其中 reduction=0 即一级邻居能量完全清零，二阶邻居能量不变。reduction=0.5 即一阶邻居能量缩减为原来的 50%，二阶邻居变为原来的 75%，reduction=1 即一阶邻居能量不做缩减，二阶邻居能量完全清零。从图 5-1 和图 5-2 可以看出，取 reduction=0.5 时，KLSER 算法在两个网络中的影响范围都不错，故在下面的涉及到 reduction 取值的实验

中，均取 reduction=0.5，即能量做半缩减。

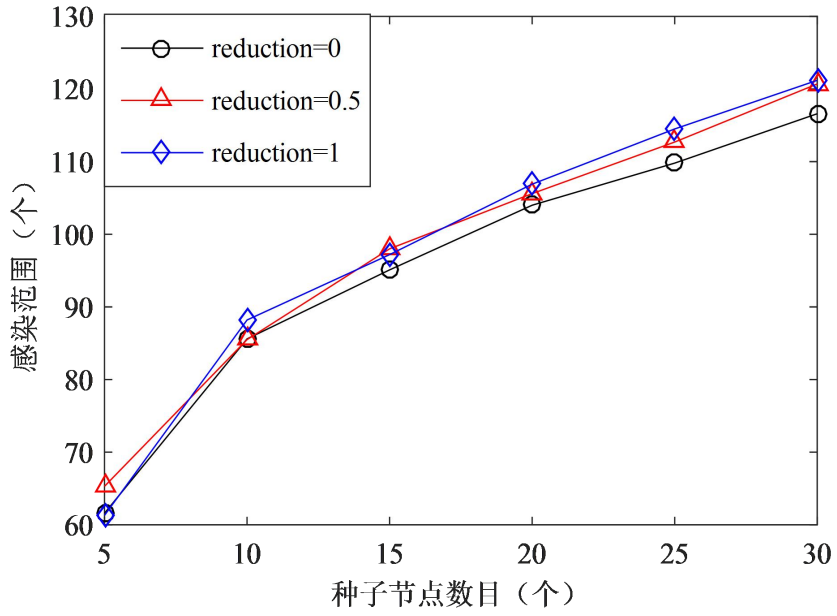


图 5-1 Jazz 上 reduction 取值对感染范围的影响

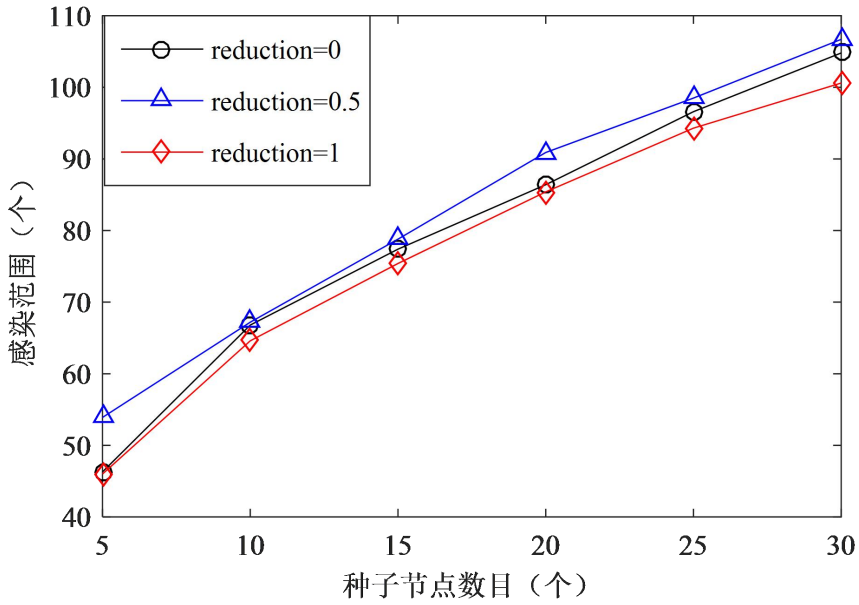


图 5-2 Neural 上 reduction 取值对感染范围的影响

在仿真过程中，感染阈值在很大程度上影响了感染的范围。当感染阈值 β 较低时，也就是人们对信息的普遍接受度较高时，每一个节点都能将它大部分的邻居感染成功，这样一传十、十传百的进行下去，消息便很快的在网络中蔓延。在这样的环境中，仅仅几个影响力大的节点就能获得很大的影响范围。随着感染阈值的升高，人与人之间的信任度不断降低，同样的种子节点集的影响范围也随之

变小。Jazz 网络上各算法影响范围随激活阈值的变化情况对比如图 5-3 所示，从图 5-3 中看出，随着激活阈值升高，KLSER 算法和 k-shell 算法影响范围都在减小，并且无论激活阈值如何变化，KLSER 算法的影响范围都明显优于 k-shell 算法。

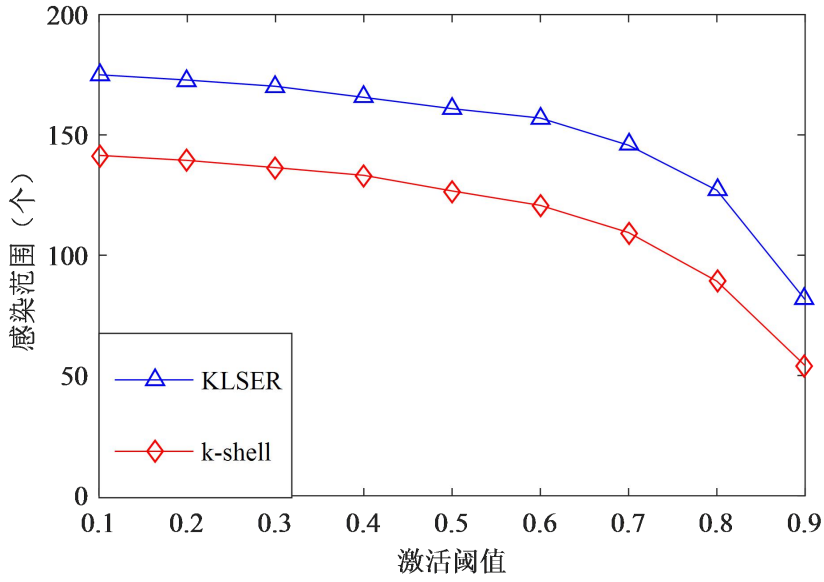


图 5-3 Jazz 上各算法影响范围随激活阈值的变化情况

图 5-4 为 Neural 网络各算法影响范围随激活阈值的变化情况对比图。由图知 Neural 网络中两个算法的影响范围都随阈值的增大而减小，并且两算法曲线重合度较高，说明在 Neural 网络上两算法的影响范围相近，但是准确来讲 KLSER 算法的影响范围略优于 k-shell 算法，并且随着感染阈值的增大，这种优势愈加明显。

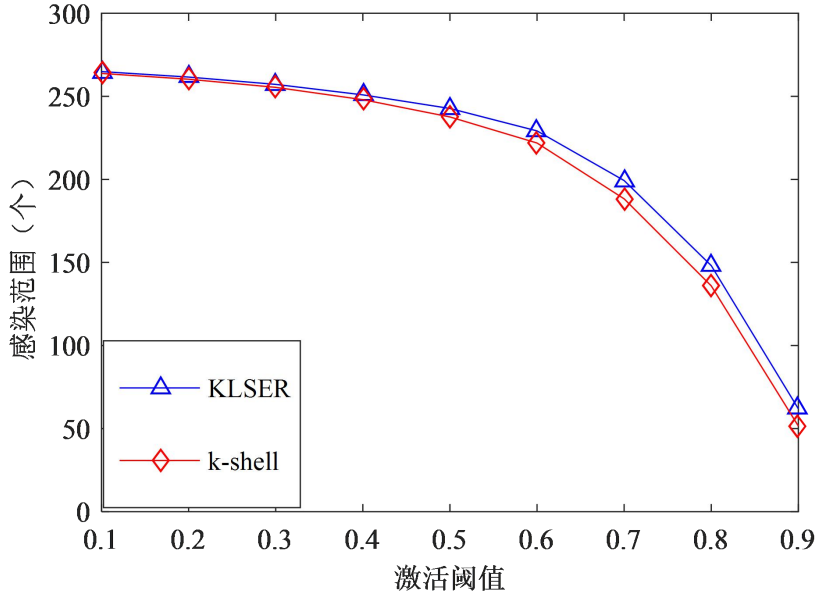


图 5-4 Neural 上各算法影响范围随激活阈值的变化情况

从上述分析可知，在激活概率较低时，网络中的节点较容易被激活，k-shell 算法以及经典的中心性算法都能达到很大的影响范围，算法优劣区分的不明显。因此，接下来在比较 KLSER 算法与其他算法的影响范围时，为了较明显的区分各个算法的优劣，均取激活阈值 β 为 0.9，也就是研究低信任度下各个算法的效果。

Jazz 网络上 KLSER 算法和 k-shell 算法种子节点集质量的对比如图 5-5 所示。从图 5-5 中可以看出，随着种子节点集规模的增大，两个算法的感染的范围也随之增大，但是 KLSER 算法感染范围增长的较明显，而 k-shell 算法感染范围增长的较缓慢。究其原因，k-shell 算法仅仅选择了网络中的核心节点，没有考虑节点间的影响力重叠问题，导致后续选入种子集的节点和种子集先前已有节点的影响范围发生重叠，后续选入种子集的节点的影响范围被削弱。而 KLSER 算法考虑到了这个因素，对种子节点集进行局部能量缩减的优化，大大降低了种子节点间影响力的重叠。

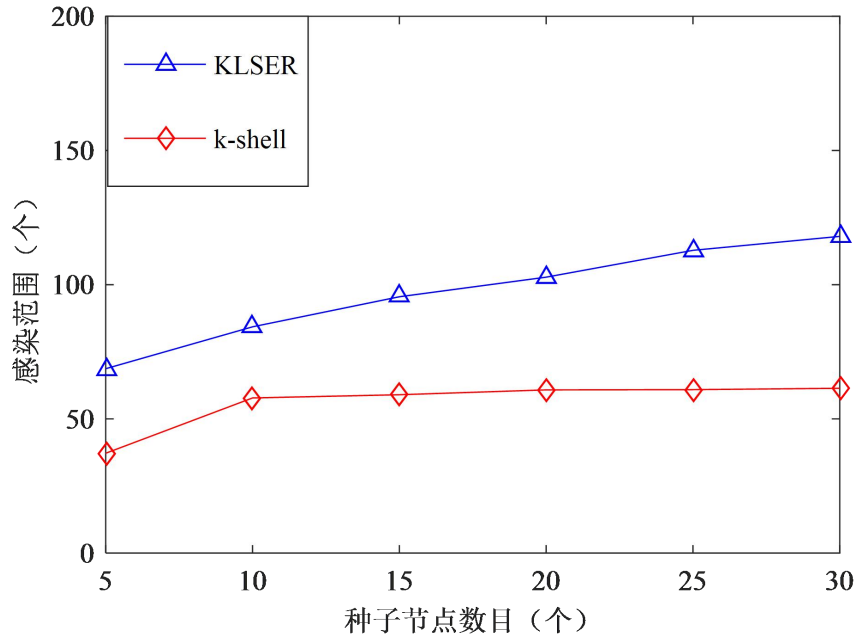


图 5-5 Jazz 上各算法种子节点集质量对比图

Neural 网络上 KLSER 算法和 k-shell 算法种子节点集质量对比如图 5-6 所示。观察图 5-6 可知，在种子节点集规模小于 10 个时，两算法影响范围较接近，随着种子节点集规模的增大，两算法影响范围的差距愈加明显并且 KLSER 算法的影响范围优于 k-shell 算法。分析原因，主要是 KLSER 算法考虑了种子节点间的影响范围重叠问题，因此随着种子集合规模的增大，KLSER 算法的优势更加明显。

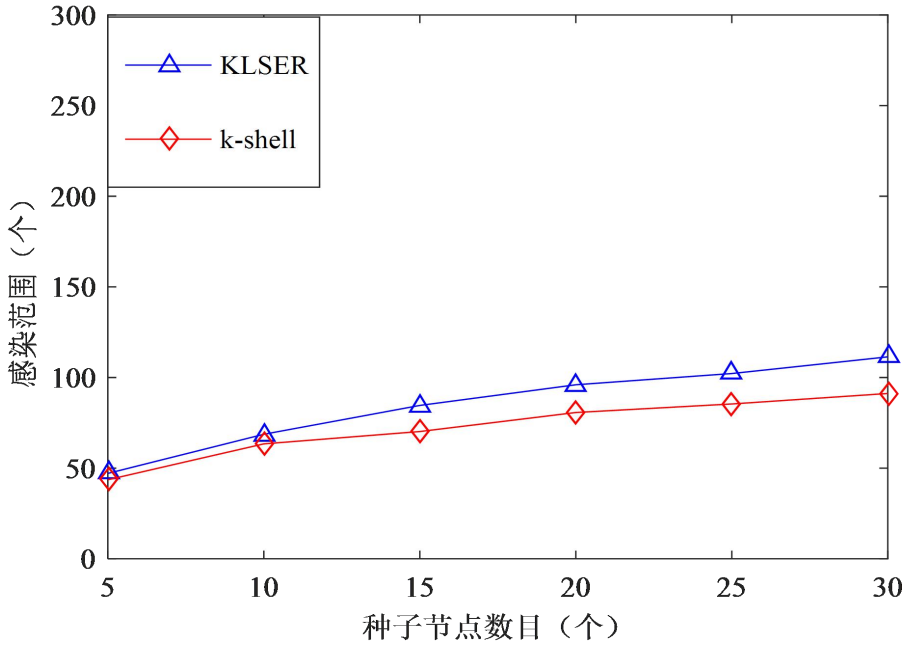


图 5-6 Neural 上各算法种子节点集质量对比图

目前为止，本章通过实验证明了在 Jazz 真实网络数据集和 Neural 真实网络数据集上，本文第 3 章提出的基于 k-shell 的影响力最大化 KLSER 算法选出的种子节点集合在影响范围上略优于传统的 k-shell 算法选出的种子节点集合。

接下来，本章将第 3 章提出的基于 k-shell 的影响力最大化 KLSER 算法与影响力最大化研究领域内 3 种经典的中心性算法：度中心性算法、紧密中心性算法、介数中心性算法进行对比。

第 3 章提出的基于 k-shell 的影响力最大化 KLSER 算法与经典的 3 种中心性算法的种子节点集质量对比如图 5-7 所示。观察图 5-7 明显可知，四个算法的影响范围由大到小排序分别为：KLSER 算法、度中心性算法、介数中心性算法和紧密中心性算法。其中 KLSER 算法与 Degree 度中心性算法影响范围较为接近，但是 KLSER 算法略优于 Degree 度中心性算法。Closeness 紧密中心性算法与 Betweeness 介数中心性算法影响范围较为接近，但 Betweeness 介数中心性算法略优于 Closeness 紧密中心性算法。综合来看以上四种算法，第 3 章提出的基于 k-shell 的 KLSER 算法在影响范围上优于其他三种经典的中心性算法。除此之外，从算法时间复杂度的角度来看，基于 k-shell 的影响力最大化 KLSER 算法复杂度为 $O(n^2)$，远远低于基于图的最短路径的 Closeness 紧密中心性和 Betweeness 介数中心性算法的时间复杂度。

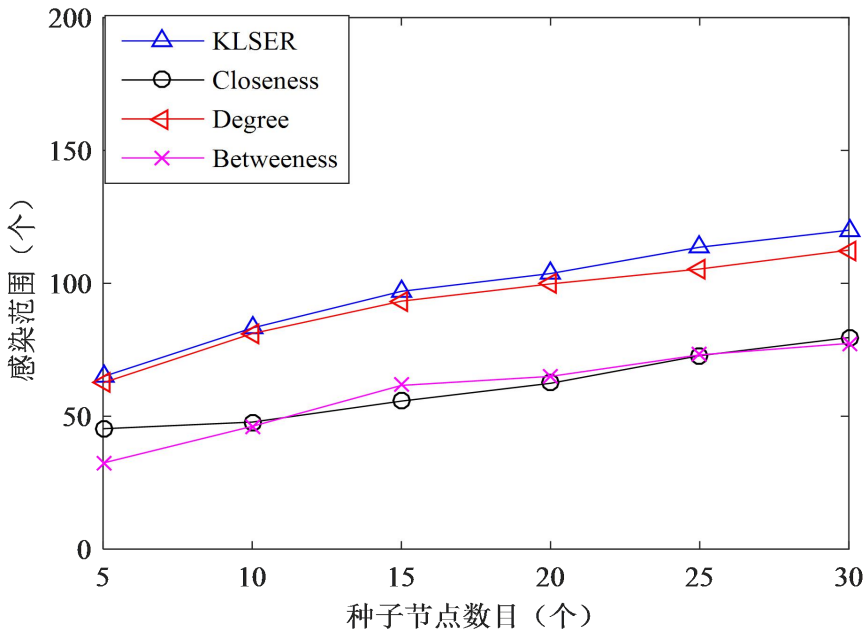


图 5-7 Jazz 上各算法种子节点集质量对比图

Neural 网络上 KLSER 算法与经典中心性算法种子节点集质量对比如图 5-8 所示。观察图 5-8 可得 4 个算法的性能排序为：KLSER 算法、紧密中心性算法、度中心性算法和介数中心性算法。虽然 4 个算法的影响范围差距较小，但 KLSER 算法在影响范围上略优于其他三种经典的中心性算法。并且从时间复杂度的角度讲，KLSER 的复杂度低于基于最短路径的紧密中心性和介数中心性算法。

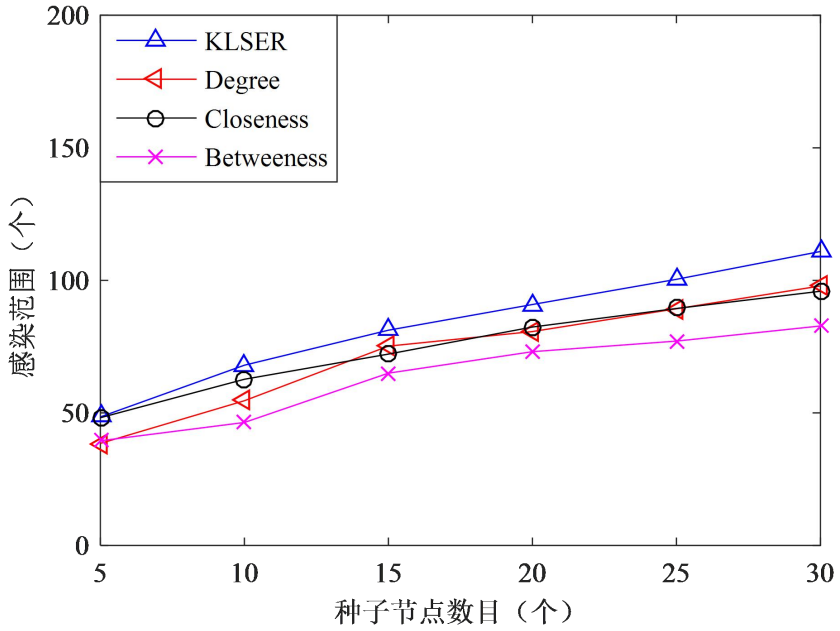


图 5-8 Neural 上各算法种子节点集质量对比图

5.4 基于社区结构的 IBC 算法实验结果及分析

5.4.1 实验数据集

E-mail network URV 是 Rovira I Virgili(Tarragona)大学邮件网络[42]。网络中的节点表示该大学里某个研究小组的成员，边代表小组内成员之间互通邮件的关系。该网络包含 1133 个节点和 5451 条边，是一个无向无权网络。

Wiki-Vote 网络是 Wikipedia 的投票历史网络，其中节点代表 Wikipedia 的用户，u 到 v 的有向边意味着 u 投票给了 v，该网络共有 7115 个节点，103689 条边，是一个有向无权网络。两个网络的基本信息如表 5-2 所示。

表 5-2 两个网络的基本信息

序号	数据集	特点	节点数	边数	平均度
1	E-mail network URV	无向无权图	1 133	5 451	9.62
2	Wiki-Vote	有向无权图	7 115	103 689	13.3

本次实验选取了 E-mail network 和 Wiki-Vote 这两个真实网络数据集作为测试集，首先用 Pajek 软件中的 Louvain Method 算法对两个网络进行社区划分，算法将 E-mail network 网络划分为了 10 个社区，模块度 Q=0.58，划分结果如图 5-8 所示。

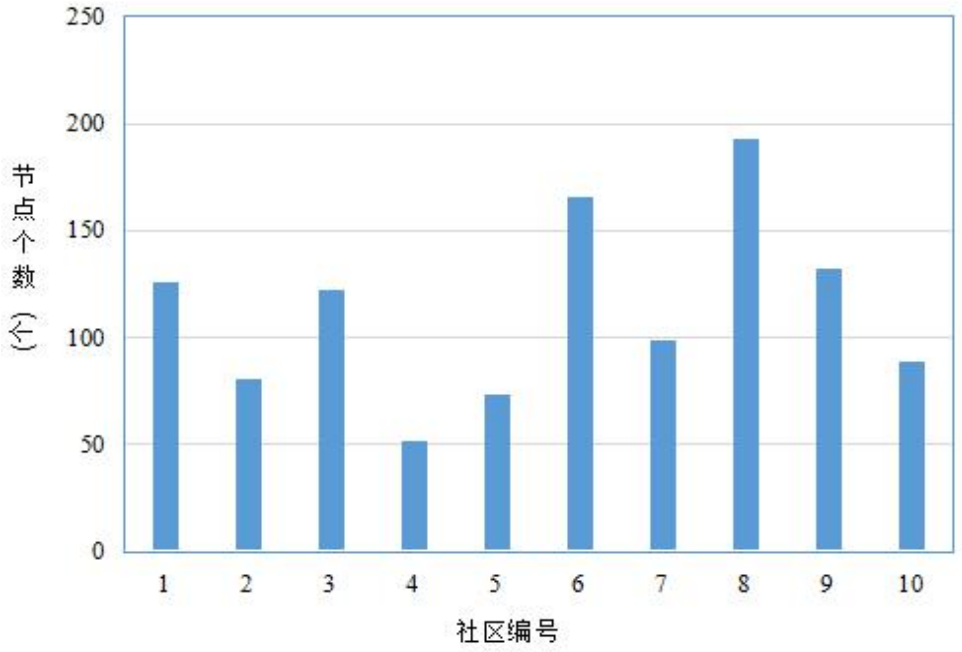


图 5-8 Email 网络社区划分结果

由图 5-8 可得，Louvain Method 算法划分的社区规模较均匀，大部分社区的节点个数都在 100 左右，只有 4 社区节点个数较少，只有 50 个。6 社区和 8 社区节点个数较多，超过 150 个。Louvain Method 算法将 Wiki-Vote 网络划分为 5 个社区，模块度为 Q=0.42，划分结果如图 5-9 所示。由图 5-9 可得，每个社区的个数在 1000~2000 之间，社区规模划分的较均匀。

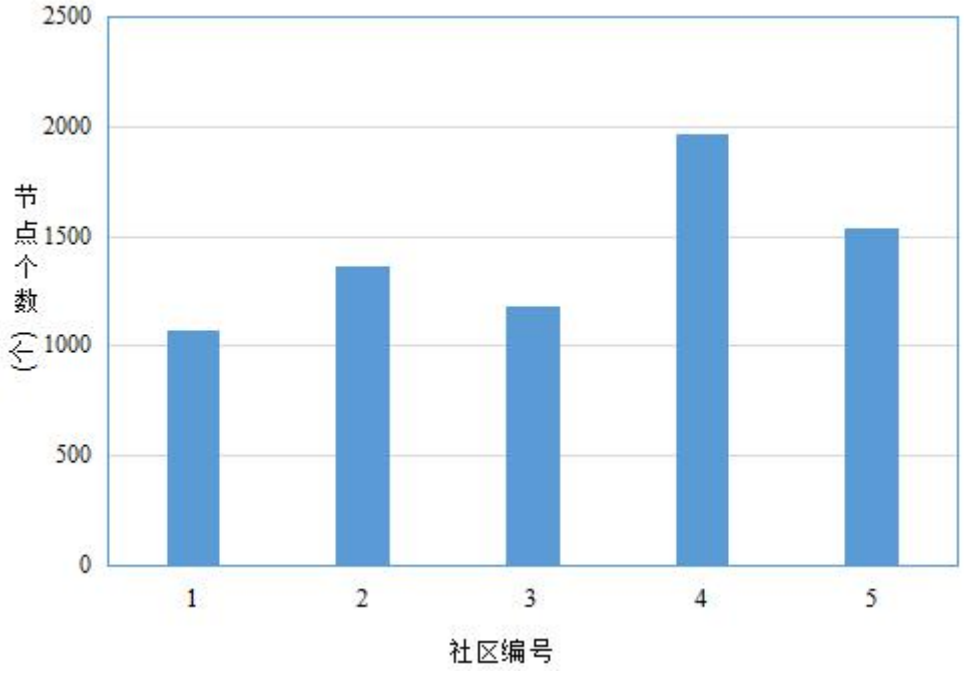


图 5-9 Wiki-Vote 网络社区划分结果

5.4.2 IBC 算法的实验结果及分析

本次实验选取了 Email network 和 Wiki-Vote 这两个数据集作为测试集，以影响力最大化领域 3 种经典的中心性算法：度中心性算法、紧密中心性算法、介数中心性算法作为对比算法，以种子节点集最终感染的节点数目做为评价指标，探究不同算法选取的种子节点集的质量。

针对每个数据集，在独立级联模型上做 100 次仿真实验取平均值。当激活阈值较低时，人们对信息的接受度较高。每个对比算法都能获得不错的影响范围。所以这里本文取激活阈值 $\beta = 0.9$，探究高激活阈值下 IBC 算法与其他算法的种子节点集质量的优劣。

本文首先要探究的是第 4 章公式(4-12)中从社区内核心节点集和整个网络边

界节点集选取节点的比例系数 $t(0<t<1)$ 的取值。每个社区的核心节点集至关重要，对该社区内信息传播起到了不可替代作用。但整个网络的边界节点的作用在于沟通各个社区，使这些个孤立的社区连成一个整体，也是必不可少的，但是仅仅少量的重要的边界节点就能连通整个网络。因此，对于整个网络，从核心节点集选取的节点数量较多，因此比例 t 应较大一些，接近于 1。

Email network 真实网络数据集和 Wiki-Vote 真实网络数据集上从社区内核心节点集和整个网络边界节点集选取节点的比例系数 t 取不同值时 IBC 算法影响范围对比如图 5-10 和图 5-11 所示。在这里，比例 t 的取值选取了 3 个有代表性的值，分别是 0.1、0.5 和 0.9。当 t 取 0.1 时，表示从社区内核心节点集选取的节点较少，从整个网络的边界节点集选取的节点较多。当 t 取 0.5 时，表示社区内核心节点集和边界节点集取节点的数量大体相同。当 t 取 0.9 时，表示主要从社区内核心节点集选取节点点进入种子集合。

综合分析图 5-10 和图 5-11 可以发现，核心节点集选取节点的比例系数 t 的取值越大，基于社区结构的 IBC 算法的影响范围也随之越大。当比例系数 t=0.9 时 Email 网络和 Wiki-Vote 两个真实网络的数据集都高于其他两条曲线，达到了最优的影响范围，故在下面的实验中，从社区内核心节点集的选取节点的比例系数 t 的取值均为 0.9。

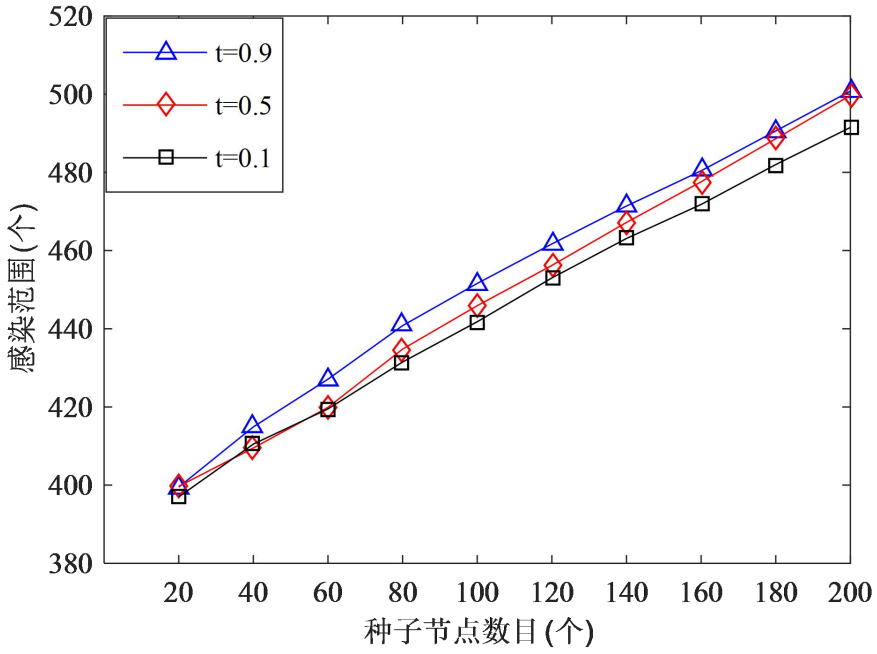


图 5-10 Email 上 t 的取值对种子节点集感染范围的影响

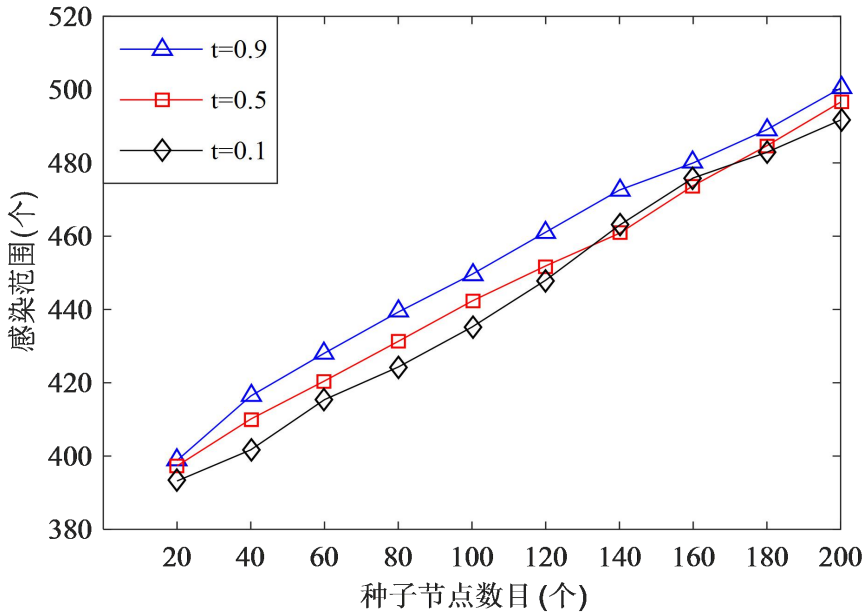


图 5-11 Wiki-Vote 上 *t* 的取值对种子节点集感染范围的影响

Email network 和 Wiki-Vote 网络上 IBC 算法与三种主要的经典中心性算法的种子节点集质量对比分别如图 5-12 和图 5-13 所示。从图 5-12 和图 5-13 可以看出，随着种子节点集规模增大，各个算法的影响范围也随之增大，但增大的不明显。分析原因，主要是因为种子节点之间的影响力重叠。并且 IBC 的影响范围要优于其他三种经典中心性算法。

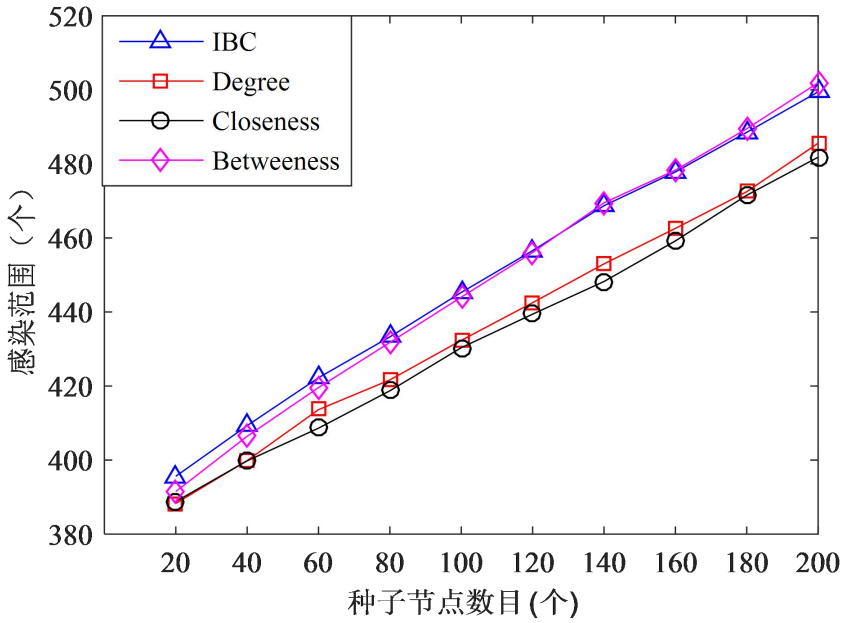


图 5-12 Email 上各算法种子节点集质量对比图

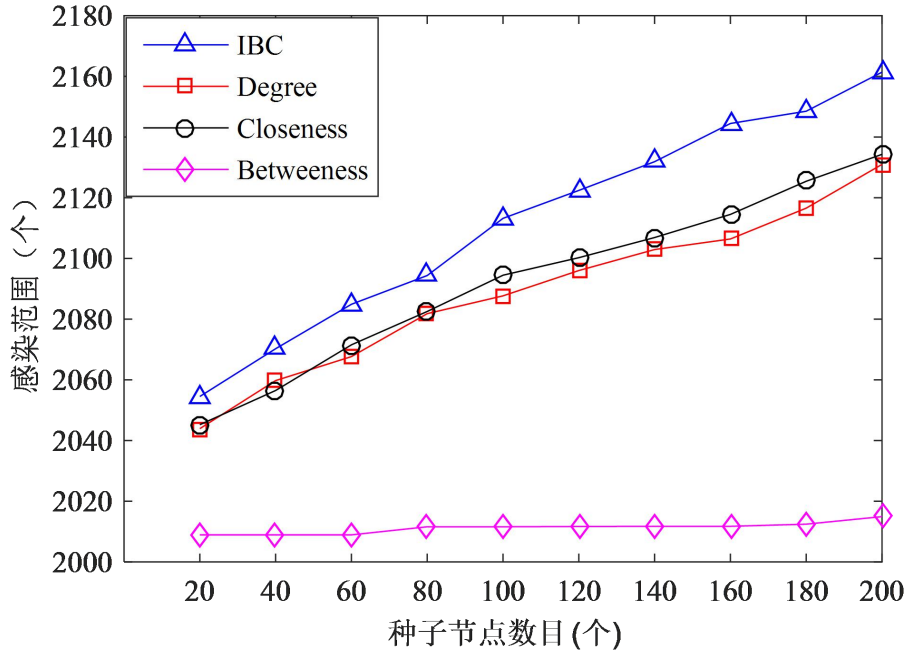


图 5-13 Wiki-Vote 上各算法种子节点集质量对比图

从图 5-12 和图 5-13 中还可以发现，同一种算法在不同的网络中表现性能也有差异。例如图 5-12 介数中心性算法表现仅次基于社区结构的 IBC 算法，但在图 5-13 中表现结果却最差，而且增加种子节点的个数对影响范围几乎没有影响。分析原因主要与该网络结构有关，导致计算得出的每个节点的介数中心性值都很小，甚至很多节点的介数值为 0。介数中心性算法在衡量该网络节点影响力时作用不明显。

接着，探究不同激活阈值下基于社区结构的 IBC 算法和其他三种经典的中心性算法的影响范围。由于 Emai 网络和 Wiki-Vote 网络规模相差较大，Email 网络有 1133 个节点，Wiki-Vote 网络有 7115 个节点。因此两个网络种子种子节点集规模设置必须不同。根据实际情况，设定 Email 网络种子节点集规模为 k_1=50，Wiki-Vote 网络种子节点集规模 k_2=200。

Email 网络激活阈值对感染范围影响仿真结果如图 5-14 所示。观察图 5-14 可知，基于社区结构的 IBC 算法和其他三种经典中心性算法的曲线在激活阈值小于 0.9 时几乎完全重合，只有在激活阈值为 0.9 时基于社区结构的 IBC 算法的影响范围才略优于其他三种经典的中心性算法。说明对于 Email 网络，本文第 4 章提出的 IBC 算法在低中激活阈值下没有显示出其优势，只有在高激活阈值即人与人信任度较低时才能发挥其优势。

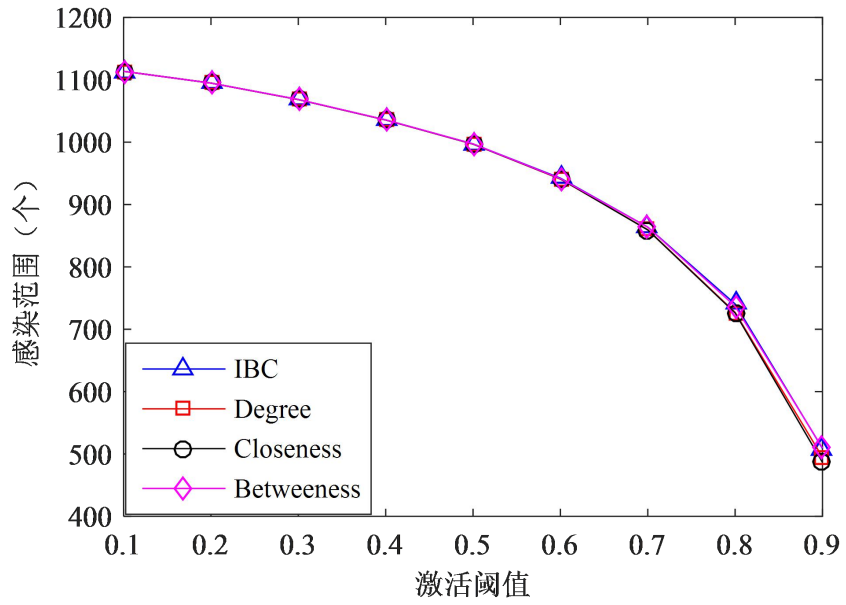


图 5-14 Email 网络上激活阈值对感染范围的影响

Wiki-Vote 真实网络数据集上激活阈值对感染范围的影响的仿真结果如图 5-15 所示。观察图 5-15 可以发现，无论激活阈值如何变化，基于社区结构的 IBC 算法的影响范围都优于其他三种经典的中心性算法(度中心性算法、介数中心性算法、紧密中心性算法)。除此之外，从算法时间复杂度角度来分析，第 4 章提出的基于社区结构的 IBC 算法复杂度低于基于网络最短路径算法的介数中心性算法和紧密中心性算法。

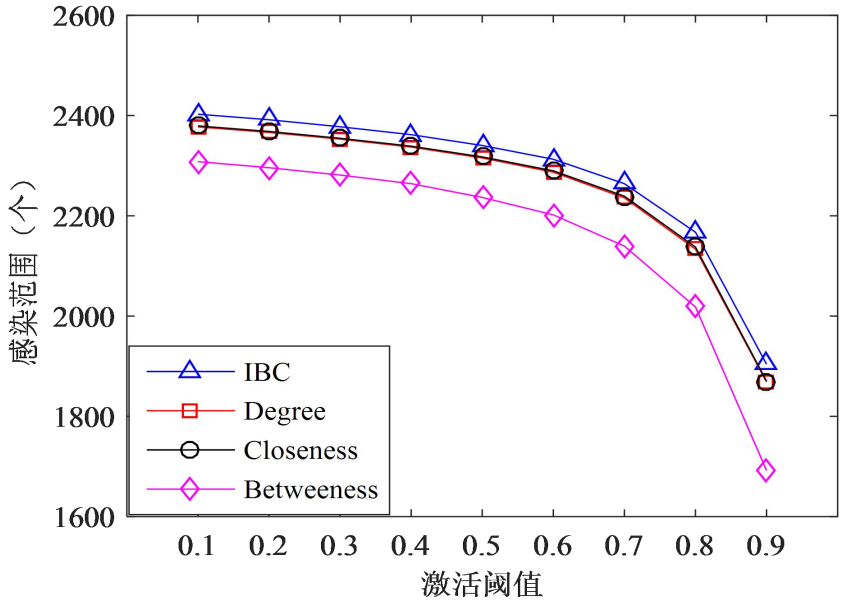


图 5-15 Wiki-Vote 网络激活阈值对感染范围的影响

5.5 本章小结

本章首先介绍了实验环境，然后介绍了仿真模型的实现及评价标准。接着用真实网络的数据集在独立级联模型上进行感染节点的仿真实验，并和经典的中心性算法对比，验证了本文提出的两个算法的准确性和有效性。除此之外，对两个算法中所涉及到的参数 *reduction* 和 *t*，都进行实验探究以确定其最优的取值。综合来看，本文提出的两个算法 KLSER 和 IBC，都达到了预期的实验效果，取得了优于传统算法的影响范围，并且复杂度都保持在 $O(n^2)$。

结　论

影响力最大化问题，即寻找网络中 k 个最关键的节点，使得这 k 个节点能够获得最大的影响范围，是一个具有重要实践价值的 NP-hard 课题。近些年来，国内外众多学者从扩大种子节点集的影响范围和降低算法的时间复杂度两方面提出了很多基于经典算法的改进算法，给该领域的研究带来了很多新的思路同时也存在着一些不足之处。本文在阅读大量的文献资料并分析经典的算法及其优秀的改进算法过程中，发现了 k-shell 算法识别网络核心的有效性和 k-shell 对网络划分的粗糙性以及社区结构对处理大规模网络的优势。基于这两点，本文做了深入的研究，并取得了一些研究成果，主要如下：

(1) 重点分析了 k-shell 算法的原理及不足，结合消息在网络上的传播机制，提出了一种基于 k-shell 算法的改进算法。综合考虑节点在网络中的位置及其邻居的贡献定义每个节点的影响力。并依据节点的单个影响力和种子节点之间的影响力重叠两方面考虑提出了基于能量缩减的种子节点集选取策略。在独立级联模型上进行了 3 个仿真实验，分别是探究能量缩减百分比 reduction 的取值的实验、探究不同算法选择的种子节点集的质量实验和探究影响激活阈值对影响范围的影响的实验。综合分析表明，本文提出的基于 k-shell 算法的改进算法有效的解决了 k-shell 算法对网络划分粗糙的问题并且该算法选出的规模为 k 的种子节点集质量优于经典的中心性算法并具有较低的时间复杂度。

(2) 深入的学习了网络的社区结构特性并提出了一种基于社区结构的影响力最大化算法，将整个网络划分成若干个社区结构。定义每个社区的核心节点集和整个网络的边界节点集。在定义边界节点的影响力时，根据两社区联系的程度和连接社区的规模来定义社区间连边的权值。并且在确定每个社区选取的种子节点比例时，认为该比例与该社区被其他社区激活的程度成反比。最后在独立级联模型上进行了两个仿真实验，分别是该算法与经典中心性算法的影响范围对比实验和该算法在不同激活阈值下的影响范围实验。综合分析表明该算法的影响范围优于传统的中心性算法并且具有较低的时间复杂度，能够适用于大规模网络。

本文虽然提出了两个新的算法，较原有的算法稍有改进，但是还有待进一步研究和扩展，主要可以进行以下几方面的工作：

本文研究的只是无权网络的影响力最大化问题，加权的网络更符合实际的应用问题。因此，在以后的研究工作中，可以将该算法扩展到加权网络中去。

在 KLSER 算法中，只考虑了节点的一阶邻居的 K_s 值和出度，还可以扩展到二级邻居及更多。另外在能量缩减策略中，也只缩减了种子节点一阶和二阶邻居。

在第 IBC 算法中，本文使用的是常用的 Louvain Method 方法划分的社区结构，还可以尝试其他划分社区结构的算法，并加以比较。

参考文献

[1] 李熙. 基于六度分割理论和中心度识别微博网络的关键人物[D]. 成都:西华大学计算机技术硕士学位论文,2013: 7-8.

[2] Newman M E J. The Structure of Scientific Collaboration Networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2000, 98(2):404-9.

[3] Jeong H, Tombor B, Albert R, et al. The Large-scale Organization of Metabolic Networks[J]. Nature, 2000, 407(6804):651-654.

[4] Latora V, Marchiori M. Is The Boston Subway a Small-world Network[J]. Physica A Statistical Mechanics & Its Applications, 2002, 314(1):109-113.

[5] Williams R J, Martinez N D. Simple Rules Yield Complex Food Webs[J]. Nature, 2000, 404(6774):180-3.

[6] Redner S. How Popular is Your Paper? An Empirical Study of the Citation Distribution[J]. The European Physical Journal B, 1998, 4(2):131-134.

[7] Adamic L A, Lukose R M, Puniyani A R, et al. Search in Power-law Networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2001, 64(2):046135.

[8] Amaral L A N, Scala A, Barthélémy M, et al. Classes of Small-world Networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2000, 97(21):11149-11152.

[9] Richardson M, Domingos P.. Mining Knowledge-sharing Sites for Viral Marketing[C]// ACM SIGKDD international Conference on Knowledge Discovery and Data Mining.New York ACM, 2002:61-70.

[10] 刘丽丽. 网络产品病毒式营销研究[D]. 北京:对外经济贸易大学企业管理硕士学位论文,2006: 1-8.

[11] Xu J, Chen H. Untangling Criminal Networks: A Case Study[C]// Intelligence and Security Informatics, First NSF/NIJ Symposium, ISI 2003, Tucson, AZ, USA, June 2-3, 2003, Proceedings. DBLP, 2003:232-248.

[12] 何俊. 基于复杂网络理论的电力系统连锁故障分析[D]. 吉林:东北电力大学电气工程硕士学位论文, 2013: 16-18.

[13] Kempe D. Maximizing the Spread of Influence in a Social Network[J]. Progressive Research, 2008:137-146.

[14] Leskovec J, Krause A, Guestrin C, et al. Cost-effective Outbreak Detection in Networks[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.New York ACM, 2007:420-429.

[15] 田家堂, 王轶彤, 冯小军. 一种新型的社会网络影响最大化算法[J]. 计算机学报, 2011, 34(10): 1956-1965.

[16] Chen W, Wang Y, Yang S. Efficient Influence Maximization in Social Networks[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July. DBLP, 2009:199-208.

[17] Liu X, Li M, Li S, et al. IMGPU: GPU-Accelerated Influence Maximization in Large-Scale Social Networks[J]. IEEE Transactions on Parallel & Distributed Systems, 2014, 25(1):136-145.

[18] Diestel R. Graph theory[J]. Mathematical Gazette, 2011, 173(502):67-128.

[19] Freeman L C, Borgatti S P, White D R. Centrality in Valued Graphs: A Measure of Betweenness Based on Network Flow[J]. Social Networks, 1991, 13(2):141-154.

[20] Chen D, Lü L, Shang M S, et al. Identifying Influential Nodes in Complex Networks[J]. Physica A Statistical Mechanics & Its Applications, 2012, 391(4):1777-1787.

[21] 陈勇, 胡爱群, 胡啸. 通信网中节点重要性的评价方法[J]. 通信学报, 2004, 25(8):129-134.

[22] Restrepo J G, Ott E, Hunt B R. Characterizing The Dynamical Importance of Network Nodes and Links.[J]. Physical Review Letters, 2006, 97(9):094102.

[23] 刘耀庭. 社交网络结构研究[D]. 杭州: 浙江大学计算机应用硕士学位论文,2008:41-45.

[24] Cha M, Haddi H, Benevenuto F, et al. Measuring User Influence in Twitter[J]. 2010(2):148-152.

[25] Weng J, Lim E P, Jiang J, et al. TwitterRank: Finding Topic-sensitive Influential Twitterers[J]. Wsdm, 2010:261-270.

[26] Kitsak M, Gallos L K, Havlin S, et al. Identification of Influential Spreaders in Complex Networks[J]. Nature Physics, 2011, 6(11):888-893.

[27] Liu J G, Ren Z M, Guo Q. Ranking The Spreading Influence in Complex Networks[J].

Physica A Statistical Mechanics & Its Applications, 2014, 392(18):4154-4159.

[28] 顾亦然, 王兵, 孟繁荣. 一种基于 K-Shell 的复杂网络重要节点发现算法[J]. 计算机技术与发展, 2015(9):70-74.

[29] Bae J, Kim S. Identifying and Ranking Influential Spreaders in Complex Networks By Neighborhood Coreness[J]. Physica A Statistical Mechanics & Its Applications, 2014, 395(4):549-559.

[30] Zeng A, Zhang C J. Ranking Spreaders by Decomposing Complex Networks[J]. Physics Letters A, 2012, 377(14):1031-1035.

[31] Hu Q, Gao Y, Ma P, et al. A New Approach to Identify Influential Spreaders in Complex Networks[J]. 2013, 62(14):99-104.

[32] 张成弼. 复杂网络中关键节点的发现研究[D]. 西安:西安电子科技大学硕士学位论文, 2013: 35-36.

[33] Galstyan A, Musoyan V, Cohen P. Maximizing Influence Propagation in Networks With Community Structure[J]. Phys Rev E Stat Nonlin Soft Matter Phys, 2009, 79(5 Pt 2):056102.

[34] 冀进朝, 黄岚, 王喆,等. 一种新的基于社区结构的影响最大化方法[J]. 吉林大学学报理学版, 2011, 49(1):93-97.

[35] Cao T, Wu X, Wang S, et al. OASNET: An Optimal Allocation Approach to Influence Maximization in Modular Social Networks[C]// ACM Symposium on Applied Computing. DBLP. Sierre, Switzerland, 2010:1088-1094.

[36] 郭进时, 汤红波, 吴凯,等. 基于社区结构的影响力最大化算法[J]. 计算机应用, 2013, 33(9):2436-2439.

[37] Barabási A, Albert R. Emergence of Scaling in Random Networks[J]. Science, 1999, 286(5439):509-512.

[38] Barabási A L, Albert R, Jeong H. Mean-field Theory for Scale-free Random Networks[J]. Physica A Statistical Mechanics & Its Applications, 1999, 272(1–2):173–187.

[39] Martin T, Zhang X, Newman M E. Localization and Centrality in Networks[J]. Physical Review E, 2014, 90(5-1):052808-052808.

[40] Kitsak M, Gallos L K, Havlin S, et al. Identification of Influential Spreaders in Complex networks[J]. Nature Physics, 2011, 6(11):888-893.

[41] Pablo M, Danon L,. Community Structure In Jazz[J]. Advances in Complex Systems, 2011,

06(4):565-573.

[42] Guimerà R, Danon L, Díazguilera A, et al. Self-similar Community Structure in a Network of Human Interactions.[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2003, 68(2):065103.

致 谢

时光荏苒，岁月如梭。三年的硕士生涯已接进尾声。这三年的时光既漫长又短暂，其中既充满了酸甜苦辣，又有收获和成长。借此机会，感谢陪我一起度过美好时光的每位尊敬的老师和亲爱的同学，正是你们的帮助，我才能克服困难，正是你们的指导，我才能解决疑惑，衷心的谢谢你们。

感谢我的导师刘永山教授，在他的悉心指导和帮助下，本论文才能顺利完成。从论文的选题，资料的收集，到论文的修改，到最后的定稿。刘老师给予我许多宝贵的意见和耐心的指导。刘老师开阔的视野、严谨的治学态度、精益求精的工作作风，深深地感染和激励着我。此外，刘老师还为我提供了燕山大学软件中心的宝贵实习机会，通过实践锻炼提高专业技能。在此，向尊敬的刘老师表示衷心的感谢和由衷的敬意，祝您在以后的工作和生活中万事如意，心想事成！

感谢孔德瀚博士师兄，韩贵春博士师姐和实验室其他成员在实验过程和论文写作过程中提供的热心帮助!无论在炎热的夏天，还是寒冷的冬季，你们不辞劳苦地为我提供无私的帮助，没有你们的帮助就没有这篇论文的顺利完成。

感谢我的父母和家人多年来的支持,你们让我更加乐观地面对生活中的一切，你们对我如此无私的付出，我会用一生去回报。

感谢燕山大学信息科学与工程学院对于的教育，燕大永远是我的后盾，无论到哪我都会谨记我出自燕大，要给燕大争脸。也感谢所有帮助过我的行政老师们。

最后，我要向百忙之中参与审阅、评议本论文各位老师、向参与本人论文答辩的各位老师表示由衷的感谢!人生的每个阶段都值得好好珍惜，这段美好岁月，因为有你们的关心和帮助，我很幸福。我会更加勤奋学习、认真研究。把最美好的祝福献给你们，愿永远健康、快乐！