分类号：_____ 密级：_____

UDC：_____ 编号：_____

工学硕士学位论文

社会网络影响力最大化算法研究

硕士研究生：王丽婷

指导教师：董宇欣　副教授

学科、专业：计算机应用技术

论文主审人：董红斌　教授

哈尔滨工程大学

2014 年 3 月

分类号：_____ 密级：_____

UDC：_____ 编号：_____

工学硕士学位论文

社会网络影响力最大化算法研究

硕士研究生：王丽婷

指导教师：董宇欣　副教授

学位级别：工学硕士

学科、专业：计算机应用技术

所在单位：计算机科学与技术学院

论文提交日期：2014 年 1 月

论文答辩日期：2014 年 3 月

学位授予单位：哈尔滨工程大学

Classified Index:

U.D.C:

A Dissertation for the Degree of M. Eng

Research of Influence Maximization Algorithms in Social Networks

Candidate:	Wang Liting
Supervisor:	A.Prof. Dong Yuxin
Academic Degree Applied for:	Master of Engineering
Specialty:	Computer Applied Technology
Date of Submission:	January,2014
Date of Oral Examination:	March, 2014
University:	Harbin Engineering University

哈尔滨工程大学
学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献的引用已在文中指出，并与参考文献相对应。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者（签字）：

日期：　　年　月　日

哈尔滨工程大学
学位论文授权使用声明

本人完全了解学校保护知识产权的有关规定，即研究生在校攻读学位期间论文工作的知识产权属于哈尔滨工程大学。哈尔滨工程大学有权保留并向国家有关部门或机构送交论文的复印件。本人允许哈尔滨工程大学将论文的部分或全部内容编入有关数据库进行检索，可采用影印、缩印或扫描等复制手段保存和汇编本学位论文，可以公布论文的全部内容。同时本人保证毕业后结合学位论文研究课题再撰写的论文一律注明作者第一署名单位为哈尔滨工程大学。涉密学位论文待解密后适用本声明。

本论文（□在授予学位后即可　□在授予学位12个月后　□解密后）由哈尔滨工程大学送交有关部门进行保存、汇编等。

作者（签字）：　　导师（签字）：

日期：　　年　月　日　　年　月　日

摘 要

互联网技术的发展已经在逐渐的改变人们的生活，社交网络的兴起使得人与人之间的联系更加方便，同时一些商家借助于网络，利用一些名人的“口碑效应”进行某产品的推广，使得该产品的影响达到最大化，这就是社会网络影响力传播最大化问题比较简单的定义。在2002年Domingos等人首次对影响力最大化问题给出比较详细的定义，在这以后影响力最大化问题逐渐成为社会网络领域的一大研究热点。学者们结合社会网络的理论分析，并根据真实网络中的传播规律，建立了各种社会网络的传播模型，并结合各种传播模型提出了各种影响力最大化的算法，用于解决社会网络中的影响力最大化问题。

本文详细研究了社会网络中已有的一些影响力最大化算法及传播模型之后，针对影响力最大化算法中存在的效率和效果问题提出了一种并行社区搜索影响力最大化算法，并针对独立级联模型中激活概率不变的问题提出一种改进传播模型。本文的主要研究工作包括以下两个方面：（1）提出一种社会网络影响力最大化算法，即并行社区搜索影响力最大化算法，并搭建Hadoop环境来做实验，验证该算法可以应用于大规模社会网络，在社区上查找最具影响力节点，可以扩大影响力传播范围，然而将这种查找实现并行化，提高了算法运行的性能；（2）提出一种浮动传播级联模型，该模型随着激活节点数量的增减不断的修正激活概率使其比较符合实际的传播规律，并通过做对比实验，证明了该模型更符合实际的传播规律。

关键字： 社会网络；影响力最大化；并行社区搜索；传播模型

Abstract

The development of Internet technology has been gradually changing people's life, the rise of social networks makes the connections between people more convenient. At the same time, some businesses take advantage of "word-of-mouth" of celebrities to make their products well known, this is the influence maximization problem. Influence maximization problem is first proposed by Domingos ,which has become a hot research topic in the field of social networks. Scholars combined the theory of social network analysis, the transmission of all kinds of social network model were established according to the propagation rules of real network. In addition, the various combination of propagation model influence maximization algorithms were proposed, which be used to solve the influence maximization problem in social networks.

In this paper, we propagation a parallel search community influence maximization and an improved propagation model based on the independent cascade model. This paper studies the following question:(1) The parallel search community influence maximization algorithm is proposed to solve the problems of social networks influence maximization algorithm, and we build a Hadoop environment to do the experiments. The final results show that the algorithm can be applied to large scale social network. Search the most influential nodes can expand influence spread range in the community. The search for the realization of parallel which could improve the performance of running the algorithm;(2) We propose an improved propagation model based on the independent cascade model. The model constantly revises the activation probability in the information dissemination process to make it more in line with the actual communication process and it is proved by experiments that results of the propagation in this model is more in line with the reality results.

Keywords: social networks; influence maximization; parallel search community; propagation model

目　　录

第1章 绪论

1.1 研究背景与意义

互联网技术的发展，带来了各种社交网络（Facebook，Twitter，人人网等）的蓬勃发展，现在它已成为人们在线交流的主要交流工具，通过网络人们可以结交朋友、共享信息。网络中这些丰富的信息数据给知识发现和数据挖掘带来了前所未有的挑战。目前社会网络中的社区划分、影响力、信息传播模型等研究内容成为研究热点。社交网络中人与人之间在交往过程中会出现各种各样的关系，比如：朋友关系、同学关系、同事关系等等。人们之间的这些关系之间会相互影响，这种影响作用有大有小，因此本文从这种相互影响的角度来重点研究影响力最大化的问题。

影响力最大化问题最早由 Domingos 等人[1]提出，主要应用于对“口碑效应（Word-of-Mouth）”和“病毒式营销（Viral marketing）”的研究。“病毒式营销”是指营销开始时，在网络中寻找小部分有“影响力”的成员，给他们发放一些该产品的样品，让他们免费试用，再利用这些最初选出的“影响力”成员的口碑效应引发一连串的影响，这些“影响力”成员会将这些产品推荐给他们的朋友，朋友又会将新产品推荐给他们的朋友，这一连串的推荐就像病毒一样往外蔓延。这种营销模式有一个突出的优点，那就是人们比较容易接受来自关系亲密的人的推荐，而广告营销模式就缺乏这种优点，这种方式可以给商家带来丰厚的回报。但是我们应该如何选有“影响力”的成员，最终让更多的人知道并购买我们的产品，是我们应该考虑的主要问题。

社会网络中的信息时刻在传播着，有时候我们需要促进这些信息的传播，而有时候我们需要抑制这些信息的传播。比如在“病毒式营销”模式中，为了促进商品的传播，我们需要选择最具“影响力”的用户作为传播源头，从而让更多的人了解或购买我们的产品。但是现实中有些疾病在社会群体中流行，网络中谣言的流传，计算机病毒在网络上的传播等这些不良传播，这些都需要我们采取有效措施，从源头扼制这不良传播，使得传播达到最小化。网络中的这些传播实际上都遵循着某种传播规律，通过对这些现象的分析，我们可以找出它们之间的传播规律，也就能很开的找到传播源头，这对促进或制约事物的传播行为具有重要意义。

1.2 国内外研究现状

近些年，学术界对社会网络影响力最大化问题研究颇多，学者们针对该问题提出了很多这方面的算法。根据社会网络影响力求解方法不同，可以分为基于全局传播的影响力最大化算法和基于社区传播的影响力最大算法，以下根据这两个方面分别介绍关于该问题的国内外研究现状。

1.2.1 全局网络影响最大化研究现状

在 2002 年 Domingos 等人[1] 最早开始影响力最大化问题的研究，该问题提出之后受到广大研究者的关注。在该文章中作者对影响力最大化问题给出了详细定义，为后来的学者研究影响力最大化问题奠定了基础。

Kempe 等人[2]详细研究了影响力最大化问题，文中详细描述了影响力最大化问题的三个模型，即独立级联（Independent Cascade, IC）模型、线性阈值（Linear Threshold, LT）模型和权重级联（Weight Cascade, WC）模型，并证明了影响力最大化问题是 NP 难题。同时他们还提出了一个应用到三个模型上的贪心算法，该算法保证了影响力传播范围具有（1-1/e）（e 为自然对数的底数）最优解传播范围，但是贪心算法有存在一个明显的缺陷，当网络规模达到一定数量，贪心算法的时间复杂度会特别高，因此在大规模网络中使用贪心算法是不可行的，针对这个问题后来的很多学者给出了一些对贪心算法的改进算法，甚至提出了新的其他算法，使其在规模较大的网络中仍然能获得较好的效率。

Leskovec 等人[3]提出一种新的影响力最大化节点的优化方案，作者为该方案命名为“Cost-Effective Lazy Forward”（CELF）算法。CELF 算法运用影响力最大化问题的子模特性，明显降低了节点传播影响的次数，通过实验证明，使用 CELF 算法选择的种子节点，其传播效果可提高 700 倍。

Chen 等人[4]对贪心算法进行改进之后提出两种影响力最大化算法，称为 NewGreedy 算法。NewGreedy 算法是将网络中没有影响力的节点从原始网络中移除掉，得到一个比原始网络规模更小的网络，然后从规模较小的网络中进行传播影响，这样在每次模拟传播中不就需要从整个网络考虑，提高了算法的效率。同时作者还改进了独立级联模型，将其和 NewGreedy 算法合并称之为 MixGreedy 算法，MixGreedy 算法第一步用 NewGreedy 算法选择第一个种子节点，第二步用 CELF 算法选取影响力最大的节点，通过对比实验，作者提出的算法与贪心算法相比，其时间效率要高效的多。

虽然 Chen 等人改进的贪心算法在时间复杂度上有所降低，但是该算法仍然不适用

于大规模社会网络，与此同时，他们又提出了 DegreeDiscount 算法[4]，该算法是对选取度最大节点算法的一种改进，实验结果表明，DegreeDiscount 算法的效果比将节点度数排序，选择值最大的节点作为种子节点的启发式算法有改进。

Kimura and Saito[5]提出基于最短路径的影响力模型，并在此模型下提出了一个寻找最大影响力节点的高效算法，与一般的传播级联模型不同的是，他们并没有改进贪心算法的效率问题。

Narayanam 和 Narahari 提出基于合作博弈沙普利值解概念的 SPIN 算法[6]，该算法主要分两步进行，1）遍历整个网络，计算每个节点的沙普利值；2）从列表中选取 k 个沙普利值最大的节点添加到种子集合中。该算法相比于其他算法，其效率有显著提高。

田家堂等人[7]利用线性阈值模型“影响力累积”的特性并综合考虑网络的结构特性和传播特性，提出了一种影响最大化框架，并在此框架下提出一种新型的混合影响最大化算法 HPG。HPG 算法分两个阶段：启发阶段和贪心阶段。首先在启发阶段选取 PI 值最大的节点，其次在贪心阶段利用贪心算法选取影响力最大的节点。实验结果表明 HPG 算法无论是在传播效果上还是在运行时间上都比贪心算法获得较好的效果。

陈浩等人[8]针对线性阈值模型，并综合考虑了节点之间的影响力和节点的激活阈值，提出了一种基于节点激活阈值的启发式算法，该算法分两个阶段：启发阶段和贪心阶段，在启发阶段根据节点动态变化的激活阈值来更新 PIN 值，每次都选取 PIN 值最大的节点，在贪心阶段根据贪心算法选取影响力最大的节点。实验结果表明该启发式算法和 HPG 算法相比，该启发式算法的传播范围更广。

1.2.2 社区网络影响最大化研究现状

上一节提到的大部分算法都没有考虑网络的社区结构，Girvan 和 Newman 在 2002 年提出了网络的社区概念，并提出了检测网络社区结构的 GN 算法[9]。之后的学者们提出了很多社区发现算法，这些算法所研究的主要内容包括社区发现、重叠社区发现及动态网络中的社区发现等等[8-10]。2007 年，Scripps.Jerry 和 Pang-Ning Tan[10]首次在他们的文章中提到了社区影响的概念。2009 年，A.Galstyan 和 V.Musoyan[11]等人第一次将社区影响的概念应用到解决影响力最大化问题中。该作者使用的主要方法是将整个网络划分为两个社区，然后在两个社区上寻找影响力最大的节点，该算法虽然降低了算法的时间复杂度，但是该算法仅仅适用于由两个松散社区组成的随机网络，因为现实网络中不一定只存在两个社区，所以该算法具有一定的局限性。

CaoTianyu 等作者[12]将社会网络的影响力最大化问题看作是资源的最佳分配问题，并在社区划分的基础上提出了 OASNET 算法。该算法的思想是首先采用现有的社区发现算法将社会网络划分成若干个独立的社区结构，然后采用动态规划的思想将种子节点最佳的分配到各个社区上，最后在各个社区上采用某种传播模型做影响力最大化算法实验，并将在各个社区上得到的影响力最大的节点做累加起来得到最终被影响到节点的数目，最后作者还证明了使用该方法来求解影响力最大化问题也是一个 NP 难题。

WangYu 等人[13]也提出自己的基于社区划分的思想求解影响力最大化问题的 CGA（Community-base Greedy algorithm）算法。该算法的基本思想是首先采用社区划分算法将整个网络划分为不同的社区结构，然后采用动态规划的方法在各个社区上查找最具影响力的 k 个节点。实验结果表明该算法在精度上有少量损失，但是该算法与之前提到的算法相比，在时间复杂度上仍然很高。

周夏冰等人[14]从并行的角度提出一种社会网络影响力传播的并行算法 CPCGA。CPCGA 算法通过社区划分，定义社区的计算量，采用递减贪心算法来实时保证计算节点的负载均衡来提高效率。实验结果表明，CPCGA 算法在保证精度的情况下，较大的提高了算法的效率。

以上详细的讲述了目前社会网络中影响力最大化问题的研究现状。Kempe 等人[2]提出的贪心算法及后人在贪心算法的基础上提出了很多改进的贪心算法[4,5]在时间复杂度上都特别高，因此这些算法根本不适用于大规模社会网络中。现实中很多社会网络大都存在社区结构[15]。在社区内部节点之间联系十分密切而社区之间的联系十分松散，也就是说，网络中的每个节点与其所在社区内部的节点交互比较频繁，而受其他社区上的节点的交互比较稀疏，基于这种社区划分的思想求解影响力最大化问题是可行的。目前利用社区发现的思想求解影响力最大化问题还是比较少的，而且目前已有的算法大都忽略了社区之间的联系，这是不符合实际的，而且这种将社区孤立的算法会影响传播的最终效果。本文从社区划分的角度并考虑社区之间的联系及社区之间的天然并行性，首先利用现有的快速社区发现算法[16]将整个网络进行社区划分，并保留社区之间的联系，最后采用并行策略在各个社区内寻找影响力最大的节点。

1.3 论文主要研究内容

本文的研究内容如下：

第一，阐述该课题的研究背景与研究意义，并对目前影响力最大化问题的研究现状

给出了详细阐述，最后对该领域现存的问题进行本文的研究工作。

第二，阐述社会网络相关理论及两个比较经典的影响力传播模型：独立级联模型（IC）和线性阈值模型（LT）并重点研究四种主要的影响力最大化算法，包括贪心算法、Degree discount 算法、CGA 算法和 CDH-Kcut 算法，分析各算法的优缺点。

第三，根据这些算法的不足提出了一种改进算法，即基于社区划分的并行挖掘影响力最大化节点算法（PC-NIE），该算法首先通过 BGLL 算法将网络划分为小于 k 个的不同社区，其次将 k 个初始节点按比例分配给不同的社区，在各个社区上并行计算每个节点的度数和节点邻居度数之和的加权来得到种子节点集合，最后通过进行实验对比来验证本文提出算法的有效性。

第四，针对独立级联模型中存在的传播概率在传播过程中不变的问题，提出一种改进的传播模型（FTC），并通过实验验证该模型的有效性。

1.4 论文组织结构及内容安排

本文主要针对社会网络中的影响力最大化问题，提出一种影响力最大化算法，并通过分析影响力传播模型，提出一种改进的影响力传播模型。

全文共分为五章，各部分内如简述如下：

第 1 章，绪论部分主要阐述课题研究背景和研究意义、该领域国内外研究现状以及存在的问题，最后简要概述了本文的研究工作。

第 2 章，介绍社会网络相关理论及几种目前常用的影响力传播模型，详细介绍了两个经典的影响力传播模型，即独立级联模型（IC）和线性阈值模型（LT）。

第 3 章，主要详细讲述了四个典型的影响力最大化算法，并总结算法中存在的优缺点。

第 4 章，根据第三章算法中存在的问题及不足，提出了一种基于社区划分的并行挖掘社区上影响力最大的节点算法，给出了算法的思想、算法的描述及算法分析，最后通过搭建 Hadoop 实验环境，并在真实的网络中进行实验，最后比较详细的分析了实验结果。

第 5 章，首先针对第二章中介绍的独立级联模型中存在的问题，提出了一种改进的传播模型，其次对实验用到的数据集给出了简要介绍及进行实验结果对比，最后比较详细的分析了实验结果。

第2章 社会网络相关理论及传播模型

本章主要介绍社会网络的基本概念和社会网络传播模型的相关理论知识，重点介绍主要的社会网络影响力传播模型。

2.1 相关概念

2.1.1 社会网络

图是由顶点的集合和边的集合组成，其中边将两个不同顶点相连在一起（并且两个顶点之间至多只有一条边）。早在 1736 年网络就作为一种被称为“图”的数学对象得到关注，当时伟大的数学家欧拉研究了著名的哥尼斯堡七桥问题。在普鲁士的城市哥尼斯堡有七座桥，欧拉问道，是否可以有一条散步的路线，走过所有这七座桥，而没有一座桥走过两次。他证明了这是不可能的，这就是图论的第一个定理（Euler 定理）。由此诞生了图论、拓扑学。

一个社会网络[17]是由单个群体或团体按照某种关系连接起来而构成的系统。在这个系统中存在各种各样的关系，比如人与人之间的这种朋友关系、同事与同事之间的这种合作关系、家庭与家庭之间的这种联姻关系和公司与公司之间的这种商业关系等等。社会网络抽象为图（Graph），可以符号化表述为：$G=(V,E)$，其中 $V=\{v_1,v_2,\ldots,v_n\}$，$E=\{e_1,e_2,\ldots,e_n\}$。节点的集合用符号 V 表示，包含了一组 $v_1,v_2,\ldots,v_n$ 节点，边的集合用符号 E 表示，包含了一组 $e_1,e_2,\ldots,e_n$。节点之间必须成对出现才有意义。一对节点我们用的符号是(v_1,v_2)，它们之间的连线是$<v_1,v_2>$（如图 2.1）。节点可以代表一个个体、一个公司或者一个团队。边则代表节点之间的关系。

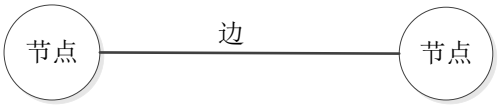

图 2.1　节点与边的关系图

“六度分割理论”[18]是社会网络中比较著名的理论，20 世纪 60 年，著名的社会心理学家斯坦利•米格拉姆（Stanley Milgram）设计了一个创造性的消息传递机制。他把一些信件散发给几百个从波士顿（Boston）、奥马哈（Omaha）、内布拉斯加州（Nebraska）

随机找的人，这些信都是发给马塞诸塞州（Massachusetts）莎朗（Sharon）地区的一个股票经纪人，他住在波士顿。但是这些信的传递要遵守以下特殊的规则。发信人必须发给自己熟悉的、知道名字的人。当然，如果他认识所说的这位股票经纪人，就可以直接寄给他。但是如果他不认识这位先生，那他就应当把信寄给自己认识，而且他认为会比较接近目标收信者的一个人。当米格拉姆问这些人，估计要通过多少次转手才能使这封信到达这位股票经纪人手中的时候，然而结果接近于六！这个就是“六度分割”的来源。“六度分割理论”说明了“弱纽带”在社交网络中是普遍存在的，这种存在使人与人之间的距离看起来非常“近”。

社会网络可以看成是由点的集合和边的集合构成的网络图，那么它也有一些类似于图的性质，描述如下：

1．节点的可达性和连通性

节点的可达性[19]（reachability）指的是网络中与该节点有路相连通的节点数量。节点的可达性描述了从该节点到达网络中的其他节点的难易度。可达性强的节点对信息的传播、资源的扩散有着举足轻重的作用。

给定一个网络G，如果G中每个顶点都可以通过某条路径到达网络中的其他节点，那么此图G为连通网络。社会网络分析法[20]中也给出了一些关于网络连通性的概念。衡量网络连通性的常用指标[19]包括，为破坏网络中节点的连通性，应使删除网络中节点或者边的数目达到最少，或者是使网络中连通元组的数目或规模最小。对于存在桥[19]（即删除导致网路不连通的边）的网络而言，网络中桥的个数也常常被作为衡量网络连通的定量指标。此外，有些文献也采用网络密度、网络直径及最大连通子图规模等度量，这在一定程度上也定量的刻画了网络的连通性。

2．节点的中心性及度分度

节点的中心性是度量网络中节点影响力的重要指标之一。在社会网络分析法[20]中被称之为中心性分析。常用的节点中心性定义有六种[22]。

（1）度中心性（degree centrality）：度中心性定义为节点的中心性反映在节点的度数上，即连接该节点的边的数目。度中心性的表达式为

$$DC_i = \sum_{j \in N(i)} A_{ij} \tag{2-1}$$

式（2-1）表明节点中心性取值越大，节点与其他节点通信的能力越强。在现实网络中，一个节点的影响力或者重要性通常可以用节点的度数来衡量。在团队中具有很高地位的人往往拥有很多朋友或同事。

（2）接近中心性[21]（closeness centrality）：接近中心性指的是一个节点在网络中的中心性程度，经常是采用距离作为基础来衡量。如果一个节点与网络中其他节点之间的最短路径之和越小，则节点在网络中的地位越重要。接近中心性的表达式为

$$CC_i = \frac{N-1}{\sum_{1 \le j \le N} d_{ij}} \tag{2-2}$$

接近中心性在实际应用中研究的很少，因为此指标的要求很高，必须是完全连通图（fully connected graph）才能计算接近中心性。而现实中的网路可能存在非连通网络，这样的节点对之间的最短距离可令其为无穷大，从而使接近中心性趋近于零。另外，研究发现度中心性高的节点其接近中心性也高，也就是说接近中心性与度中心性有很高的相关性。

（3）中介中心性（betweenness centrality）：中介中心性是指在网络中占据重要位置的人，也就是说占据网络中重要位置的人，如果他拒绝做中介者，则两个人就中断了联系，一个人或者一个组织作为中介者的能力往往就采用中介中心性来衡量。如果一个人的中介性越高，说明他占据重要位置越多。用公式描述如下：

$$C_B(n_i) = \sum_{j<k} \frac{g_{jk}(i)}{g_{jk}} \tag{2-3}$$

其中$g_{jk}(i)$表示节点i和节点j之间通过节点i的最短路径的数量。

在有向网络中，需要考虑路径的方向。中介中心性是社会网络分析中的一个重要概念。比如，罗纳德•伯特（Ronald Burt）的结构洞理论，在该理论中就采用中介中心性来衡量一个节点是否为桥节点（brige）。这里给出桥的定义是指，处在两个非连通子图中间连接点上的节点。在社会网络中之所以重视桥的概念，就是因为处于桥节点位置的人掌握了两个分离的团体之间的信息流动和商业机会，两个团体若要信息交流、意见沟通、行动要协调等都要经过这个人—桥节点来做[21]。

3．节点的最短路径和聚集系数

最短路径（shortest path）在描述社会网络内部结构方面起着重要的作用。所谓最短路径是指：从网络中的节点v出发至另一节点u可能存在多条路径相通，在这些路径中采取什么方法可以找到一条花费最小（边上的权重和最小）的路径。一般在社会网络分析中比较常见的度量是平均最短路径（mean geodesic distance）。用公式描述如下：

$$l = \frac{1}{N(N-1)/2} \sum_{i \ge j} d_{ij} \tag{2-4}$$

其中N表示网络中节点的个数，d_{ij}表示从节点i到节点j的最短路径的数量。式（2-4）存在一个问题，即当网络中的两个节点不可达时，则它们之间的最短路径的距离被定义为∞，导致l发散。因此Newman[22]将平均最短路径改进后定义为：

$$l = (\frac{1}{N(N-1)/2}\sum_{i>j} d_{ij}^{-1})^{-1} \tag{2-5}$$

现实社会网络中，虽然节点和边的数量庞大，但是网络中连接任意两个节点之间的最短路径的平均值和最大值（网络的直径）都很小，现实网络中的平均最短路径大都呈现很小的值。例如 Fortune1000 公司董事网络为 4.6[23]；IMD 电影演员合作网络的平均最短路径长度在 3.48 和 4.54 之间[24]；NCSTRL 计算机科学家合作网络、MATH 数学家、SPIRES 高能物理学家的平均最短路径分别为 9.7，9.5 和 4[26]；Kiel 大学 Email 联网网络为 4.95[26]；软件公司竞争网络为 2.3[27]；印度汽车零配件制造供应链的最短路径为 1.596[28]。

聚集系数（clustering coefficient）是衡量网络传递的一个度量指标，主要用来描述网络中节点之间的紧密程度。通俗的讲，在网络中节点邻居的邻居也可能是该节点的邻居。聚集系数的常用公式是由 Watts 和 Strogatz[24]提出的，在该公式中先要定义节点的局部聚集系数 C_i：用网络中包含节点 i 的三角形的数目除以以节点 i 为中心连通三个节点的数目，其公式如下：

$$C_i = \frac{2E_i}{k_i(k_i-1)} \tag{2-6}$$

其中 E_i 是节点 i 的 k_i 个邻居之间连接的边的数目。所以网络的聚集系数可定义为：

$$C^{(2)} = \frac{1}{N}\sum_i C_i \tag{2-7}$$

在对社会网络理论的研究中，若要研究社区结构、重叠社区发现或者网络影响力分析及传播等，首先需要对社会网络的基本性质有个大致了解，因此上述性质对社会网络的研究有重大意义。

2.1.2 社区结构

随着学者们对社会网络的深入研究发现，真实网络中大都存在一些社区结构[18]。这种社区结构有一个相通的特点，那就是单个社区内部的个体之间的信息交流比较频繁，而社区与社区之间信息的流通比较稀少。如图 2.2 所示，是一个中学生朋友关系网，图中每种颜色代表一个社团，整个网络包含 4 个社团，在社团内部学生之间交流十分频繁，

而社团与社团之间的交流就比较稀少。

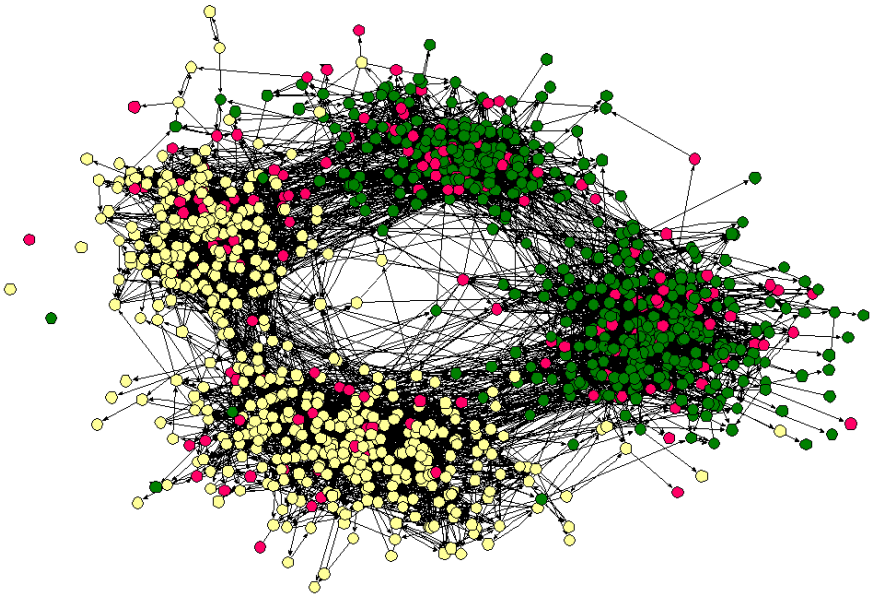

图 2.2　一个中学生朋友关系网

因为社区内部的节点之间的信息交流比较频繁，而社区与社区之间信息的交流比较稀少，所以信息在社区内部流动比在社区之间流动的快。社区内部交流的信息有可能是从社区外部传播进来的，所以社区之间也会保持有一定的联系。随着对网络性质的深入研究，人们发现许多实际网络都具有较为明显的社区结构，比如目前比较火热的人人网中每个学校就是一个社区，朋友网，天涯论坛等等。信息在这些社区内部传播的速度远比在社区外分散的节点之间传播的速度要快的多。因此在社区上研究影响力最大化问题有着重大意义。

目前关于社团检测算法已有很多学者在研究。其研究主要在两个方向上：

（1）基于模块度指标的改进算法。模块度（Modularity）是近年常用的一种衡量社团划分质量的标准[9]，其基本的思想是把划分社团后的网络与相应的零模型（Null model）进行比较，以度量社团划分的质量。所谓零模型就是指与该网络具有某些相同的性质（如相同的边数或者相同的度分布）。文献[29]在通过对模块度问题的优化分析提出了一种改进的社团发现算法，但是该算法不能将小规模社团检测出来，因此该算法具有一定的局限性。文献[30]将模块度扩展到加权网络中，对于有向网络，其方向性概率由节点的出度和入度决定，因此文献[31]将模块度扩展到有向网络中，以往的模块度定义并不适用于具有负权重的网络，文献[32]将负相关性或者不相容性引入到社团定义中，负相关性是指相互连接的两个节点完全不具备相似性；双重相关性是指正相关与负相关的混

合，即混合相关性。文献[32]还提出一种基于混合相关性的社区划分方法(DAMM)，该算法将边的正权重和负权重考虑进来，从而扩展了社区划分算法所适用的范围，最后作者还提出一个测量社团划分精度的方法，即公共重叠法。

（2）层次性社团和重叠社团划分算法。网络中的节点呈树状分布，在大的社团内可能包含一些小社团，以此类推，形成层次化的结构，这种结构的社团称为层次性社团。文献[33]是最早提出关于社团划分的层次性聚类算法的。在一个不具有层次化的网络中经过聚类得到层次化的结果，称之为层次性聚类算法。这种划分算法可能会丢掉在社团中占据重要位置的节点，也有可能使得某些节点不能正确的划分。之后文献[34]提出了一种自上而下的分裂算法，该算法可以区别出两种社会网络，一种是不具有社团特性的网络，另一种是具有社团特性但不具有层次特性的网络。但是现实中的社会网络规模都比较大，该算法在处理大规模网络时并不适用。

在基于层次的社团发现算法中，比较好的算法是文献[16]提出的快速社区发现算法（BGLL 算法），该算法的优点如下：（1）计算速度比较快，对大规模网络的社团发现比较适用。（2）该算法的执行过程是一种自下而上的凝聚过程，它可以很好的解决社团发现的分辨率问题。（3）BGLL 算法可用于加权网络中的社团发现。BGLL 算法是目前社团发现算法中比较好的算法之一，因此本文在第 4 章的社团发现部分所采用的算法就是 BGLL 算法。

2.2 影响最大化传播问题

2.2.1 影响最大化定义

对于什么是影响最大化，我们可以通过一个例子来理解。某公司开发了一款具有潜在市场价值的新产品，他们希望将该产品投放到市场中，该公司通常的做法是首先选择少数的目标用户，让他们免费试用该产品。让免费使用该产品的用户能够推荐的他们身边的朋友，朋友的朋友等等，通过这种众口相传的效果，使得尽可能多的用户能够购买该产品。所以影响力最大化问题的关键是如何选择有“影响力”的用户，通过他们的影响力使得产品的利润最大化，也就是产品影响力最大化。这个问题由 Domingos 和 Richardson 首次提出，可以大致概括如下：在一个社会网络中，已知一个信息传播模型及目标节点的个数，采用哪种方式来选择目标节点，使这些目标节点按照已知的信息传播模型进行信息的传播，并最终使得被这些种子节点影响到的节点的数目达到最大[36]，其符号化描述如下：

对于给定的社会网络 $G(V,E)$，其中 V 表示节点的集合，E 表示边的集合，假如给定一个正整数 k，如何从网络 G 中选择 k 个初始种子节点集合 A，满足$|A|=k$ 且 $A\subseteq V$，按照某种传播模型，使得集合 A 中的节点去影响网络 G 中的其他节点，最终使得集合 A 被影响到的节点数目达到最大，公式如下：

$$\max\{\sigma(A),|A|=k且A\subseteq V\} \tag{2-8}$$

其中，$\sigma(A)$为集合 A 最终影响的节点数目。

2.2.2 衡量标准

从上一小节给出的影响力最大化问题的定义的可知该问题的衡量标准从以下两个方面来衡量。

1．传播效果：当传播结束时，由这 k 个种子节点所影响到节点的数目。

2．算法效率：如何在最短的时间内找到 k 个初始种子节点集合，使其收益最大即最具影响力的节点集合。

衡量一个影响力最大化算法的好坏与否应从上述两个方面来评价，既要保证传播效果（由种子节点集合所影响到节点的个数尽可能的多），又要保证算法的效率（用最短的时间找到种子集合）。所以影响力最大化问题的最终目的就是在尽可能短的时间内找到 k 个种子集合，使得这 k 个种子集合在传播结束时所能影响到网络中节点的个数达到最大。因此影响力最大化算法需要兼顾传播效果和时间效率的平衡。

2.3 主要的影响力传播模型

信息在社会网络上的传播并使其影响力最大化是需要借助于一定的影响力传播模型的，下面是影响力传播模型中用到的定义：

定义 2.1 设 $G(V,E)$是一个社会网络，则 V 是社会网络的节点集合，任意 $v\in V$ 是社会网络中的节点，E 是社会网中节边的集合，任意$<u,v>\in E$ 是节点 u 与节点 v 的之间的关系，即边。

定义 2.2 对于社会网络 $G(V,E)$中的任意节点 v，与 v 直接相连并指向 v 的节点集合称为节点 v 的邻居集合，记为 $N(v)$。

定义 2.3 社会网络中节点的影响力是指节点所具有的能够影响其邻居节点的能力。在有向图中节点是单向传播的；在无向图中节点的影响力是双向传播的，即互为邻居的节点之间相互影响。

在社会网络的传播模型中一个节点可以用两种状态来表示，即活跃（Active）和不活跃（Inactive），活跃状态是指成功接受新产品或推荐的节点，反之为是不活跃状态，受活跃状态节点 v 的影响，不活跃状态节点 u 转变为活跃状态的过程称为节点的影响过程，一个节点一旦转为活跃状态则它就一直保持活跃状态，但不再具备影响力了，也就是说已经影响过其他不活跃节点的活跃节点不会重复影响的。在社会网络中如果一个节点有越来越多的邻居节点被影响，那么该节点被影响的概率就越大，直到某一时刻该节点被成功影响，在下一时刻该节点又去影响它的邻居节点，这一过程不断重复，直到网络中不再有节点被影响。

为了使问题简单化，假设上述传播过程中一个节点 v 的邻居节点以任意顺序尝试影响节点 v，我们称之为顺序无关性[37]。顺序无关性的定义是：一个处于不活跃状态的节点 v 的所有邻居节点对节点 v 尝试激活，但是它们对节点 v 尝试激活的顺序并不影响节点 v 最终变成活跃状态的概率。

在影响力传播模型中，比较常见的是独立级联（Independent Cascade）模型[37]和线性阈值（Linear Threshold）模型[38]，下面给出这两个模型的详细算法描述。

2.3.1 独立级联模型

下面给出独立级联模型[36]在社会网络中的传播机制：

1．给定社会网 $G(V,E)$和初始的活跃节点集合 $A=\{v_1,v_2,\ldots,v_n\}$，$v_i \in V$，在时间 t，某一节点 v 变成活跃状态之后，下一时刻它将有机会影响其不活跃邻居节点，影响概率为 p_{vu}，该值是在传播初始时刻随机设置的常数，在传播过程中是不变的，若该值越大，则节点 u 变为活跃状态的可能性就越大。

2．如果在 t 时刻有多个 v 的邻居节点受到影响，那么它们被成功影响的几率与谁先被影响，谁后被影响无关。若节点 v 成功影响了节点 u，则在 t+1 时刻，节点 u 就转变成活跃节点，

3. 在 t+1 时刻，节点 u 将继续影响其不活跃的邻居节点，重复上述影响过程。

在上述传播过程中，需要注意一点，在时刻 t 节点 v 不管能否成功影响其邻居节点 u，那么在以后的传播中，即使节点 v 一直处于活跃状态，它也不能够影响其他任何节点，我们称这一类节点为活跃的无影响力节点。传播过程结束的标志是网络中不再有新节点被影响。

2.3.2 线性阈值模型

线性阈值模型[38]在社会网络中的传播过程如下所述：

1．在传播初始时刻为社会网络 $G(V,E)$中任意节点 v 随机赋予影响阈值 $\theta\in[0,1]$，该值代表 v 被影响的难易程度，$\theta(v)$值越大，表示节点 v 越容易受到影响，$\theta(v)$值越小，表示节点 v 越难受其他节点的影响。若节点 v 受其活跃邻居的影响大于该值时，活跃邻居节点成功影响了节点 v。

2．符号 b_{vu} 是节点 u 对节点 v 的影响权值，用式（2-9）表示节点 v 的所有活跃邻居对节点 v 的总影响力。

$$\sum_{u\in N(v)} b_{vu} \le 1 \tag{2-9}$$

3．给定社会网 $G(V,E)$和初始的活跃节点集合 $A=\{v_1,v_2,\ldots,v_n\}$，$v_i\in V$，在传播初始时刻为网络中的每个节点随机赋予影响阈值，在时刻 t，若节点 v 的所有活跃邻居节点对 v 的影响力总和大于它的影响阈值时，节点 v 被激活，用式（2-10）表示：

$$\sum_{u\in N(v)} b_{vu} \ge \theta(v) \tag{2-10}$$

4．节点 v 成功受到影响后，它将在下一个时刻开始对它的不活跃邻居节点产生影响，依次重复上述传播过程，直到网络中不在有新的节点被影响，传播过程结束。

2.3.3 其他传播模型

除了上述两种比较经典的模型之外，冀进朝等人根据节点间相互影响强度的变化，提出了完全级联传播模型[39]，该模型用一个递减函数表示活跃节点对其不活跃邻居节点的影响概率。若用 S 表示节点 v 的邻居集合 $N(v)$中已经尝试激活 v 但为成功激活的节点集合，那么活跃节点 u 对不活跃邻居节点 v 的影响概率可以表示为式（2-11）：

$$p_v(u,S) = (1+k\times\frac{|S|}{|V|})\times p_v(u) \tag{2-11}$$

其中 k 是从{-1,0,1}中随机选取的值，$|V|$是社会网络中节点的个数，$S\subseteq N(v)$，$|S|$是集合 S 中节点的个数，$p_v(u)$是节点 u 对节点 v 的初始影响概率。

由于事先不知道节点间的影响何时该增强、何时该减弱，因此作者采用随机选择的方法来进行模拟传播。在该模型中活跃节点对其邻居节点的影响是动态变化的。

Kempe 等人[2,36]也提出几种其他的传播模型，如一般阈值模型（General Linear Threshold Model）、递减级联模型（Decreasing Cascade Model）和权重级联模型(Weight

Cascade Model)。Wei 等人[40-42]提出了消极评价的独立级联模型、投票模型、基于距离的模型、传染病（SIR）模型等。

综上所述，影响力传播问题需要借助事先定义的传播模型来做影响力最大化问题的传播。

2.4 本章小结

在这一章节主要介绍了与社会网络相关的一些理论知识及社会网络中主要的影响力传播模型，第一节介绍了社会网络的相关概念，包括社会网络的一些性质及社会网络中存在的一些社团结构，并大致阐述了一些现有的社区发现算法，在现有的社区发现算法中 BGLL 算法是目前社团发现算法中比较好的算法，因此在第 4 章中的社团发现部分使用了该算法；第二节介绍社会网络影响力最大化的定义以及衡量标准；第三节介绍两种比较经典的影响力最大化传播的传播模型及其他传播模型，其中独立级联模型在第 4 章进行影响力传播时需要用到，并在第 5 章本文针对该模型存在的问题进行部分改进，所以在这一章给了该模型的详细介绍。

第3章 影响力最大化相关算法

社会网络中影响力最大化问题的关键是如何选择初始目标用户使得新产品的推广范围达到最大即影响力最大化。在本章中，主要阐述几种影响力最大化的相关算法，并指出算法中存在的问题，为本文在第 4 章提出的影响力最大化算法提供了理论依据。

3.1 算法相关理论

1．子模函数。定义任意一个函数 $f(\cdot)$，将有限集合 U 的子集映射为非负实数集，如果函数 $f(\cdot)$满足收益递减（Diminishing returns）的属性，那么称 $f(\cdot)$为子模（Submodular）函数。

收益递减是指，向集合 S（$S \subseteq T$）加入任意一个元素 v（在社会网络中该元素表示一个节点）之后所收获的边际收益大于等于向集合 T 中加入相同元素所收获的边际收益，用式（3-1）表示为：

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T) \tag{3-1}$$

子模函数是指在集合中添加一个元素并不会影响 $f(\cdot)$值降低：$f(S \cup \{v\}) \geq f(S)$，也就是说子模函数具有单调递增的特点。根据这一性质可知，存在一个集合 S（$|S|=k$），使得 $f(S)$的值达到最大。文献[43]首先设置一个空的集合，然后往这个空的集合中重复的添加元素，新加入的元素能使得集合的边际收益最大。Kempe 等人[36]在他的文章中证明了在独立级联模型和线性阈值模型中，影响力函数 $\sigma(\cdot)$是子模函数。

2．NP 难题（NP-hard problem）[44]，NP 问题是指多项式复杂程度的非确定性问题。简单地说，P 类问题是指能够在多项式时间内解决的问题；而 NP 类问题是指那些在多项式时间内未能解决的问题，比如 0-1 背包问题、旅行商问题等。NP 问题的最优解目前比较常用的办法是用近似解替代最优解。Kempe 等人[36]已经证明了对于独立级联模型和线性阈值模型都是 NP 难题。文献[12]也证明了在社区上求解影响力最大化问题也是个 NP 难题。因此对于求解影响力最大化问题，大都选用近似解的办法来替代最优解。

3.2 全局网络影响最大化算法

全局网络影响力最大化问题是指在在整个网络上来寻找最具影响力的种子节点。下面给出几个基于全局网络的影响力最大化算法的详细描述。

3.2.1 Greedy 算法

Kempe 等人[36]第一次在他们的文章中提出采用经典的贪心算法（Greedy Algorithm）来求解影响力最大化问题，并证明了该算法在 IC 模型和 LT 模型中可以提供$(1-1/e)$的近似解。其中e是自然对数。贪心算法的具体描述如算法 3.1 所示：

算法 3. 1　贪心算法

Algorithm3.1 Greedy(G,K)

Input:G,R,K

1:Initialize：$S=\varnothing$

2:for i=1 to K

3:　for each vertex $v\in S\backslash V$ do

4:　　s_v=0

5:　　for i=1 to R

6:　　s_v+=|$RanCas(S\cup\{v\})$|

7:　　end for

8:　　$S_v=S_v/R$

9:　end for

10: $S=S\{argmax_{v\in V\backslash S}\{S_v\}\}$

11:end for

Output:S

上述算法 3.1 中，$RanCas(\cdot)$表示给定的随机过程，在每次的模拟过程中添加一个节点到集合 S 中，使集合 S 的影响力达到最大（第 10 行）。也就是说，每次都选择一个增量最大的节点 v 加入到集合 S 中。为此，对于网络中每个节点 $v\notin S$，影响力传播范围 $S\cup\{v\}$是通过 R 次模拟传播过程 $RanCas(S\cup\{v\})$（第 3-9 行）得到的。计算 $RanCas(S)$ 在 $O(m)$时间内完成，其中 m 是网络的边数，因此算法 2.1 的时间复杂度是 $O(knRm)$，其中 k 表示给定的种子集合的个数，n 表示网络中节点的个数，R 表示模拟传播的次数，m 表示网络中边的数目。

虽然贪心算法求解影响力最大化问题的效果较优，但是该算法的效率特别低，当网络规模很大的时候，Chen Wei 等人在文献[4]中已经实验验证了随着网络规模的增大，贪心算法的运行时间也随之快速增长，所以该算法在小规模网络比较适用。此后很多学者针对该算法做了很多改进，如 Leskovec 等人提出的 CLEF 算法[3]，并证明了 CLEF 算法的时间效率比贪心算法高达 700 倍。Chen Wei 等人提出两种改进的贪心算法，即 NewGreedy 和 MixGreedy 算法[4]，实验证明了这两种算法都比贪心算法的时间效率要高。虽然上述这些改进的算法比贪心算法的效率高，但是其时间复杂度仍然很高，并不适用于大规模社会网络。

3.2.2 DegreeDiscount 算法

在求解社会网络影响力最大问题中，有些学者提出使用网络的拓扑性质来选择初始种子节点。比如度中心性算法、中介中心性算法、接近中心性算法等等。这些算法都比较简单，通过一个公式计算每个节点的值，给节点值进行排序就可选出初始种子节点，不需要借助特定的影响力传播模型。

Chen等人提出DegreeDiscount[4]的启发式算法，该算法和贪心算法相比，在时间效率上有很大的改进，比完全基于度的算法也有一定的改善。该算法的主要思想是：当我们考虑用基于节点的度来选择节点v作为种子节点时，如果节点v的邻居节点u已经在种子节点集合S中，我们将不再考虑边$\overrightarrow{vu}$。也就是说由于节点的邻居节点u已经在种子集合中，这时我们再选择节点v作为种子节点时将给节点v的度数打个折。DegreeDiscount算法具体描述如算法3.2所示：

算法 3. 2　Degree discount 算法

Algorithm 3.2 DegreeDiscount(G,K)

Input: (G,K)
1: Initialize $S=\Phi$
2: for each vertex $v \in V$ do
3:　compute its degree d_v
4:　$dd_v=d_v$
5:　Initialize $t_v=0$
6: end for
7: for i=1 to k do
8:　select $u=argmax\{dd_v|v \in V\backslash S\}$
9:　$S=S\cup\{u\}$
10:　for each neighbor v of u and $v \in V\backslash S$ do
11:　　$t_v=t_v+1$
12:　　$dd_v=d_v-2t_v-(d_v-t_v)t_vp$
13:　end for
14: end for
Output S

上述算法 3.2 中，d_v 表示节点 v 的度数，t_v 表示节点 v 的邻居节点中已经被选为种子节点的个数，p 表示节点 v 对其邻居节点 u 的激活概率，Chen 等人认为节点对其非邻接不活跃节点的影响很小，因此可以忽略不计。用推排序的 DegreeDiscount 算法的时间复杂度为 $O(k\log n+m)$，其中 k 表示初始种子节点集合的个数，n 表示网络中节点的个数，m 表示网络中边的数目。算法 3.2 的时间复杂度与贪心算法和作者改进的贪心算法相比都低。

3.2.3 其他算法

基于全局网络的影响力最大化算法还有很多，比如文献[45] 提出改进的 Pagerank 算法，即 SanderRank 算法；文献[46]利用混色策略选择种子节点，这些策略包含节点的度、聚集系数、平均路径长度等。这些算法相比于贪心算法在时间效率上有所改进，但是效果上不如贪心算。而贪心算法及一系列改进的贪心算法在时间效率上都不太理想。

3.3 社区影响最大化算法

社区影响最大化算法是在社区上进行影响力传播，其算法思想是：先采用社区划分算法进行社区发现，然后在每个社区上进行影响力最大化传播，其中 CGA 算法和 CDH-Kcut 算法是两种主要的基于社区划分的影响最大化算法，下面给出 CGA 算法和 CDH-Kcut 算法的详细介绍。

3.3.1 CGA 算法

CGA 算法[13]是 Wang 等人利用网络中社区的性质提出一种算法；该算法包含两部分：1）利用现有的社区划分算法将社会网络划分为不同的社区；2）利用动态规划算法从各个社区上找出种子节点集合。

CGA 算法中假设已经将网络划分为 M 个社区后，面临的主要问题是主要从哪些社区中选择影响力最大的种子节点。一种简单的方法是从每个社区中均分选择，然后将这些种子节点加在一起。但是这会带来不必要的计算量。为了确定影响力最大的种子节点集合，该作者提出一种动态规划的思想来划分社区。令 $I_{k\text{-}1}$ 表示在之前 k-1 选择的影响力节点集合，用 ΔR_m 表示第 m 个社区 C_m 的影响度最大增量。用式（3-2）表示如下：

$$\Delta R_m = \max\{R_m(I_{k-1} \cup v_j) - R_m(I_{k-1}) \,|\, v_j \in C_m\} \tag{3-2}$$

上式（3-2）中的影响度 $R_m(\cdot)$的计算是在社区 C_m 上计算而不是在整个社会网络上计算。

为了挖掘网络中第 k 个影响力节点，我们首先选择要查找所有社区上产生影响力最大的增量。令 $R[m,k]$（$m \in [1,m]$ and $k \in [1,K]$）为前 m 个社区上搜索到的第 k 个最大影响力节点的影响度。如式（3-3）所示：

$$R[m,k] = \max\{R[m-1,k], R[M,k-1] + \Delta R_m\} \tag{3-3}$$

$$R[\ m,\ 0\text{=}\quad 0R\quad [k\text{=}, \tag{3-4}$$

从 m 个社区上挖掘 k 个种子节点，选取社区的函数用 $s[m,k]$表示，$s[m,k]$的定义如下：

$$s[m,k]=\begin{cases} s[m,k] & R[m-1,k]\geq R[m,k-1]+\Delta R_m \\ & R[m-1,k]<R[m,k-1]+\Delta R_m \\ m & s[0,k]=0 \end{cases} \tag{3-5}$$

为了得到第 k 个影响力节点，CGA 算法用式（3-5）选取社区，在社区上挖掘影响力节点时使用文献[4]中的 MixedGreedy 算法，具体算法描述如算法 3.3 所示：

算法 3. 3　CGA 算法

Algorithm 3.3 CGA Algorithm

Input: $(G,K,\bar{\lambda})$
Output: I
1: $C\leftarrow$ detect communities in G
2: $M=|C|$
3: $I=I_1=I_2=...=I_M=\phi$
4:for $k=1$ to K do
5: $R[0,k]=0;s[0,k]=0$
6:end for
7:for $m=1$ to M do
8: $R[m,0]=0$
9:end for
10:for $k=1$ to K do
11: for $m=1$ to M do
12: $\Delta R_m=\max\{R_m(I\cup\{v_i\})-R_m(I)|v_i\in C_m\}$
13: $R[m,k]=\max\{R[m-1,k],R[M,k-1]+\Delta R_m\}$
14: if $R[m-1,k]\geq R[M,k-1]+\Delta R_m$ then
15: $s[m,k]=s[m-1,k]$
16: else
17: $s[m,k]=m$
18: end if
19: end for
20: $j=s[M,k]$
21: $v_{\max}=\arg\max_{v_i\in C_j}(R_j(I_j\cup\{v_i\})-R(I_j))$
22: $I_j=I_j\cup\{v_{\max}\},I=I\cup\{v_{\max}\}$
23:end for

上述算法 3.3 中 4-6 行算法时间复杂度为 $O(K)$，7-9 行时间复杂度为 $O(M)$。假设整个网络划分为 C_p 个社区，执行算法中 10-19 行所需时间为 $O(MKT_p)$，算法中 20-23 行在各个社区上挖掘影响力最大的节点如果用文献[4]中的算法所需时间复杂度为 $O(K|C_p|T_p)$ 其中 T_p 表示在社区 C_p 上计算影响度的时间，如果 Q 是使用独立级联模型模拟的次数，E_p 是社区 C_p 的边的数量，则模拟传播的时间复杂度为 $O(QE_p)$，所以 CGA 算法的时间复杂度为 $O(MKT_p+K|C_p|T_p)$。

3.3.2 CDH-Kcut 算法

CDH-Kcut 算法[47]是 Chen 等人提出的一种基于社区划分的影响力最大化算法，该算法由两阶段构成：社区划分阶段和种子节点选取阶段。在社区划分阶段，发现社会网络的社区结构，在社区内个体之间联系比较频繁，社区之间个体联系稀少。该算法中社区划分所采用的算法是 Kcut 算法[48]。在种子节点选取阶段，是基于已发现的社区结构，作者提出一种种子节点选取的一种机制。

CDH-Kcut 算法的具体描述如算法 3.4 所示：

算法 3. 4　CDH-Kcut 算法

Algorithm3.4:CDH-Kcut Algorithm

Input: (G,k,p)
Output: k seeds
1:cal $Kcut(G)$
2: $S \leftarrow \phi$ //种子集合
3:select top- k biggest communities from the communities in $Kcut(G)$
4:for each selected community SC_i do
5:　add top- p degree nodes into set SC_i^p
6:for each　SC_i^p do
7:　$S \leftarrow S \bigcup$ the most degree node in SC_i^p
8:　$I_s(t) \leftarrow$ execute HDM on G with S // $I_s(t)$ 是当前种子集合 S 激活节点的集合
9:　$IM \leftarrow |I_s(t)|$ //　IM 是种子节点的最大值
10:for each community SC_i do
11:　if size(SC_i)> $avg(\sum_{i=1}^{k} size(SC_i))$ then
12:　　add SC_i in LC // LC 是最大的社区集合
13:　　sort LC based on community size
14:　　$di \leftarrow 0$ // di 是 d_node 的下标
15:for each SC_i in LC
16:　$ai \leftarrow 2$ // ai 是 a_node 的下标
17:　while $true$ do
18:　　select the ai_th large degree node from SC_i^p as a_node
19:　　select the seed candidates s_{k-di} from S as d_node
20:　　replace the d_node with a_node in S
21:　　$I_s(t) \leftarrow$ execute HDM on G with S
22:　　if $|I_s(t)| < IM$ then
23:　　　restore the replacement in line20
24:　　　break
25:　　$IM \leftarrow |I_s(t)|$
26:　　$ai \leftarrow ai+1; di \leftarrow di+1$
27:output S

上述算法 3.4 中，首先是从已划分好的社区中选取出规模最大的 *top-k* 个社区（第

1-2 行），然后在每个社区上选取中度数最大的 *top-p* 节点（第 3-5 行）作为候选集合 *pp*(*G*)，候选集合 $PP(G)=\{SC_1^P, SC_2^P, ..., SC_k^P\}$，$SC_i^P$ 表示在第 *i* 个社区 SC_i 上的度数最大的 *top-p* 节点集合，*pp*(*G*)是在 *top-k* 社区上，每个社区得到的 *top-p* 节点集合的合并集合，在文章中 *p* 的取值为社区规模的百分之十。

最终种子节点集合的选取是从候选集合 *pp*(*G*)中选取，由于 Kcut 算法在每个 SC_i^P 中不能识别在网络中占据重要位置的节点，因此选取种子节点的策略就只能采用度最大算法了。算法中作者在每个 SC_i^P 中选取度数最大的节点集合作为基本的候选节点集合 $S=\{s_1,s_2,\ldots,s_k\}$（第 6-8 行），仅仅通过度数作为候选集合可能使得节点的传播范围比较小，因此作者通过调整候选集合来得到最后的种子节点集合，其调整策略是基于一种启发式策略，即通过往候选集合中添加一个影响力大的节点来替换掉影响力比较小的节点。通过观察，在规模较大的社区上选取种子节点可能会比在规模较小的社区上选取种子节点，使得传播效果更好，因此作者给出了选取社区规模的公式，当社区的规模大于 $avg(\sum_{i=1}^{k} size(SC_i))$时，此社区将被认为是规模较大的社区（第 10-13 行），最后作者通过在规模较大的社区上选择一个 *a_node* 来替换候选集合中的一个 *d_node*（第 17-20 行），然后应用 DHM 模型看替换前后集合的影响力范围 ，如果替换后的影响力传播更广则替换掉 *d_node*（第 21-24 行）。

文中选取完种子节点后，选取某种传播模型进行传播，在初始时刻 t_0，我们选取 *k* 个节点作为种子节点，定义集合 $S=\{s_1,s_2,\ldots,s_k\}$，为了更好的模拟真实社会网络，作者所采用的传播模型是热力扩散模型（HDM），热力扩散模型是给定社会网络 *G* 和 *S*，在热能扩散的初始时刻 t_0，假定 $f_i(t_0)=h_0$（$v_i \in S$），随着时间的流逝，热量将扩散到整个网络，在时刻 *t* 如果一个个体 v_i 的热量大于或等于某个激活阈值 θ 时，该个体将被激活。

在种子节点选取阶段，作者提出两种种子节点选取的策略：1）在同一个社区中度数大的节点要比度数小的节点影响力大；2）在网络中占据重要位置的节点更有可能将信息传播的更远。由于在 CDH-Kcut 算法中所采用的社区划分算法 Kcut 算法不能识别具有重要位置的节点，因此该算法中种子节点的选取作者采用的是第一种策略。

CDH-Kcut 算法只在规模较大的社区上选取种子节点，明显降低了网络的规模，通过在真实的网络上进行实验对比，实验证明该算法在保证不影响传播效果的前提下，其算法效率比贪心算法提高了一个数量级，但是该算法仍然是忽略了社区之间的联系，影响了最终节点的传播效果。

3.4 本章小结

在这一章中，本文先是介绍了社会网络的基本概念及性质，然后详细介绍了两种经典的影响力传播模型和几种主要的影响力最大化算法，详细描述了算法、时间复杂度分析，并对算法的优缺点做出了分析，在本章的基础上，本文将提出一种改进的影响传播最大化算法，即并行社区搜索影响力最大化算法。

第4章 影响力最大化算法研究

现有的影响力最大化算法存在传播效果和时间效率不能兼顾的问题。本章将根据现有算法存在的问题，提出一种改进的影响力最大化算法，并通过实验来验证算法的高效性。

4.1 问题提出

贪心算法虽然保证了计算结果的精确性，但是贪心算法的时间效率非常低，尤其是当网络规模很大时，其算法的运行的时间要高达十几个小时，甚至几天的时间，因此贪心算法并不适用于大规模社会网络。度中心性算法比较简单，时间复杂度比贪心算法要低的多，但是基于全局网络的度中心性算法选中的种子节点容易出现重叠邻居的现象，而且度中心性算法的传播效果比贪心算法要差的多。

CGA 算法首先将网络划分为各个社区，以降低网络的规模，然后在社区上进行影响力最大化的种子节点选取，此算法的效率得到了提高，但是该算法存在的问题是，它虽然降低了社会网络的规模，但是该算法在各个社区上查找影响力最大的种子节点时使用的是贪心算法，这样算法的时间复杂度仍然很高，而且划分成各个独立的社区后，导致社区之间的联系减少，从而影响整体的传播效果。

影响力最大化问题面临的最大挑战是网络的规模庞大及传播的复杂性带来的效率问题。尽管有些学者在未来的研究工作中提出采用云计算的思想来解决影响力最大化问题，但是并没有人提出。本文基于 MapReduce 模型，从社区划分的角度提出一种 MapReduce 环境下的并行社区搜索影响力最大化算法。

4.2 HDFS 架构及 MapReduce 模型

Hadoop[49]由两部分组成，分别是分布式文件系统（HDFS，Hadoop Distributed Filesystem）和分布式计算框架 MapReduce.其中，分布式文件系统主要用于大规模数据的分布式存储，而 MapReduce 则构建在分布式文件系统之上，对存储在分布式文件系统上的数据进行分布式计算。

4.2.1 HDFS 架构

HDFS[50]是一个具有高度容错性的分布式文件系统，适合部署在廉价的机器上。HDFS 能提供高吞吐量的数据访问，非常适合大规模数据集上的应用。

HDFS 的架构[49]如图 4.1 所示，总体上采用主从(Master/Slave)结构模型，主要有以下几个组件组成：客户端、NameNode、DataNode。

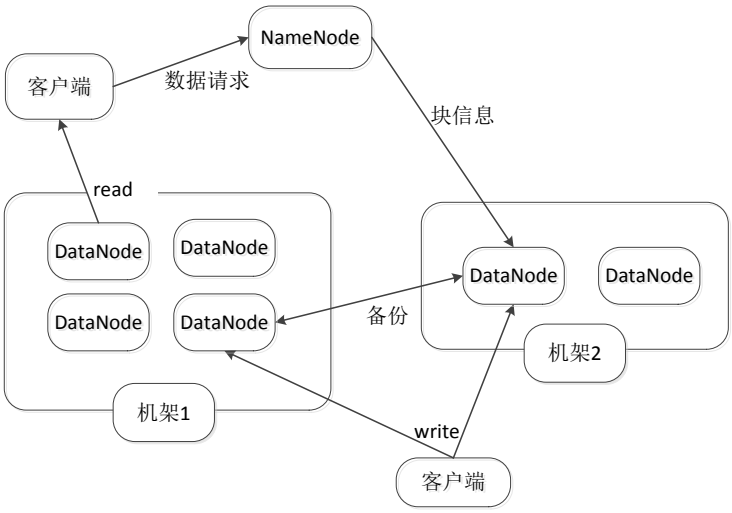

图 4. 1　HDFS 架构图

（1）客户端

客户端通过与 NameNode 和 DataNode 交互从而访问 HDFS 中的文件。客户端提供了一个类似 POSIX 的文件系统接口供用户调用。

（2）NameNode

整个 Hadoop 集群中只有一个 NameNode。它是整个系统的“总管”，负责管理 HDFS 的目录树和相关的文件元数据信息。

（3）DataNode

一般情况下，每个 Slave 节点上都安装一个 DataNode，它负责实际的数据存储，并将数据信息定期汇报给 NameNode。

4.2.2 MapReduce 模型

MapReduce 作为一种并行编程模式，使得程序编程人员这可以轻松地编写分布式并行程序。MapReduce 是基于 hadoop 平台的一种简单易用的软件框架，基于此模型 MapReduce 可以将任务发布到由上千台机器组成的集群上，并以高容错的方式并行处理海量数据，这样我们就可以实现 Hadoop 的并行任务处理功能。

图 4.2 给出了 MapReduce 作业的运行过程。从图中可以看出 MapReduce[51]执行作业需要 11 个步骤，并且涉及到 JobClient、JobTracker、TaskTracker 和 Task 4 个独立实体。JobClient 负责将用户编写的 MapReduce 程序提交到 JobTracker 端。JobTracker 对象管理所有节点的所有作业的，定义从节点的运行状态和运行方式等。 TaskTracker 对象只负责完成接受到的作业，保证节点之间的通信正常即可。如果有 Job 提交时，JobTracker 就将接收到的作业和配置信息分发给 HDFS 集群中的其他从节点，与此同时， JobTracker 还负责任务的调度以及监控 TaskTracker 的执行。

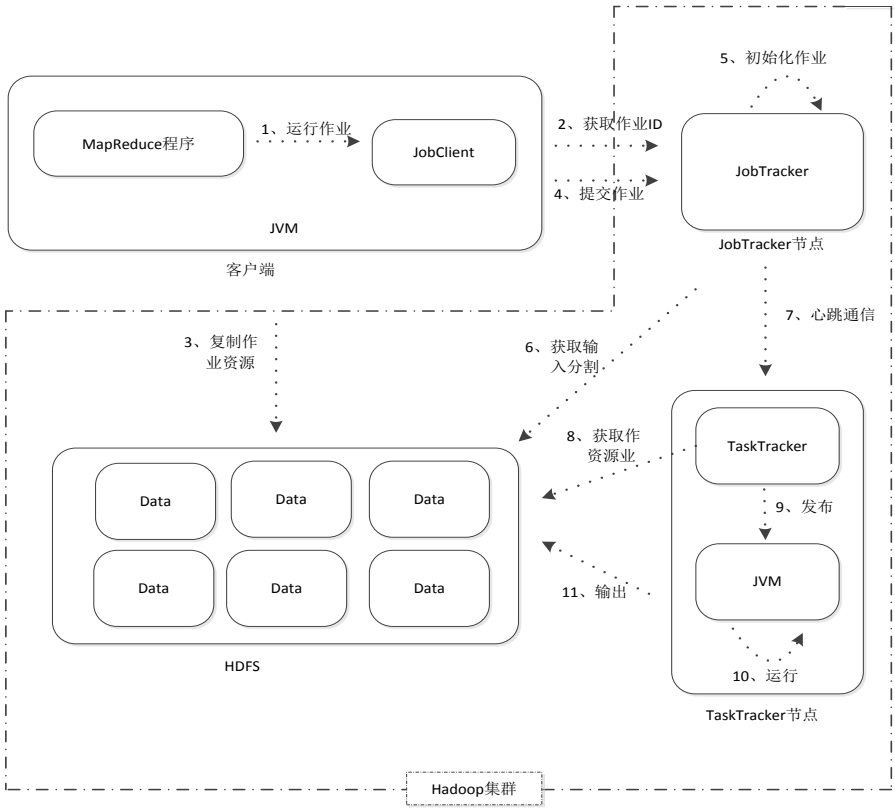

图 4. 2　MapReduce 作业的生命周期

（1）作业提交与初始化

待用户将作业提交后， JobClient 实例将与作业相关的信息上传到分布式文件系统（HDFS）上，然后告知 JobTracker，作业已提交。JobTracker 收到作业后，调用作业调度模块初始化作业。

（2）任务调度与监控

JobTracker 负责任务的调度与监控。TraskTracker 负责周期性地向 JobTracker 汇报本节点的资源使用情况，一旦有空闲资源出现，JobTracker 就按照某种策略选择一个任务使用该空闲资源，对这个任务的调度工作由任务调度器负责。另外，JobTracker 负责跟踪作业运行的整个生命周期，保证作业顺利运行完成。首先，如果 TraskTracker 或者

Task 运行作业失败，就将计算任务转移到其他 TraskTracker 或 Task 上；其次，当集群中某个 Task 执行缓慢时，JobTracker 就为其启动一个相同的 Task，最后计算结果选取计算快的 Task 计算得出的结果。

（3）任务运行环境准备

作业的运行环境需要做的准备工作包括 JVM 启动、资源的隔离，这些工作都是由 TraskTracker 来完成的，为了避免不同 Task 在运行过程中相互影响，TraskTracker 为每个 Task 启动一个独立的 JVM。与此同时，为了避免 Task 滥用资源，TraskTracker 采用操作系统进程来实现资源隔离。

（4）任务执行

TraskTracker 为 Task 准备好运行环境后就可以启动 Task 了。在作业执行过程中，每个任务及时将作业的最新进展汇报给 TraskTracker，再由 TraskTracker 汇报给 JobTracker。

（5）完成作业

当所有作业执行完成后，整个作业执的生命周期结束。

4.3 并行社区搜索影响力最大化算法（PC-NIE）

4.3.1 影响力分析

如图 4.3 所示，这是这会网络的部分片段，社会网络中的个体在该图中用节点表示，节点之间的连接表示个体之间的关系。图中节点 A 指向节点 B，表示节点 A 可以影响节点 B，而节点 B 被节点 A 影响。即在有向网络中节点指向的节点集合表示该节点可以影响到的节点集合，而指向该节点的集合表示可以影响到该节点的集合。

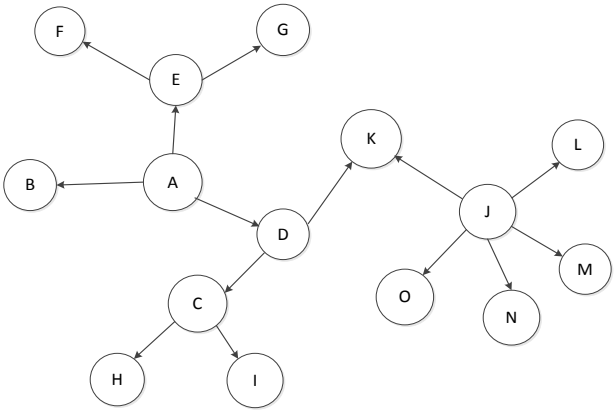

图 4. 3　社会网络部分图

节点的影响力分析情况如下：

1．节点的度

节点的度在无向图中表示节点的度数，在有向图中表示节点出度、节点的入度。有向网络中节点的出度表示个体的邻居集合，邻居越多影响到个体的可能性就越大，即影响力就越大。在图 4.3 中，节点 J 的出度是所有节点中最大的，当一个商家宣传自己的新产品时一般可以选择 J 节点作为宣传者，这是因为 J 节点和图中其他节点相比，它能直接影响到更多的节点。因此可以将节点的度数作为选择种子节点的一个可考虑的因素。

2．节点邻居的邻居集合

节点邻居的邻居集合大小有时候可以决定信息最终能够传播多远。比如在社会网络中，个体将新产品成功推荐给邻居个体，邻居再将该产品成功推荐给邻居的邻居，从而给该新产品可以带来一种像病毒式的传播，所以个体邻居的邻居数量越多，则该新产品越有可能被更多的人知道，那么该个体的影响力就会越大。

如图 4.3 中，虽然节点 J 比节点 A 能够影响更多的节点，但是通过节点 A 能够传播信息给节点 B，D，E，节点 D 又能够传播信息给节点 C 和节点 K，节点 E 能够传播信息给节点 F 和节点 G，而节点 J 将信息传播给它的邻居节点后，该信息就不能继续传播了。如果综合考虑节点的度数和节点指向的度数两个因素，则节点 A 相比节点 J 能够将信息传播到更多层的节点中。

目前已有的算法大都只考虑节点的度数这一个因素，或者将度数作为首要因素，只有度数相等的情况下才考虑节点邻居的度数和，而本文是将节点的度数和节点邻居的度数和进行线性加权，综合考虑在内。

4.3.2 算法思想描述

本算法的基本思想是：首先利用现有的社区划分算法 BGLL 算法将整个社会网络划分为不同的社区结构，利用某种数据结构保存社区划分结果，并保证社区之间的联系。如果社区划分的个数大于 k 就调整社区，将做社区的合并，最终使得社区个数小于 k。其次将种子节点个数 k 按比例分配到各个社区上。查找影响力最大的节点的方法是根据本文提出的式（4-2）。这里先将网络划分为不同的社区，使得在各个社区上查找影响力最大的前 k_i 个节点的过程提供了并行实现的可行性，因此本文提出了利用 MapReduce 模型在各个社区上并行实现查找影响力最大的前 k_i 个节点。

PC-NIE 算法的优点是将 k 个节点按照社区的规模将种子节点个数分配到各个社区中，该算法保持社区之间的联系使其不因社区之间边的损失而影响传播效果，而且该算法在社区上查找影响力最大节点时将节点的度数和节点邻居的度数和进行加权和，即式（4-2），时间复杂度比贪心算法低的多，最后该算法利用 MapReduce 模型在各个社区上并行实现查找影响力最大的节点，降低了算法的时间复杂度。

算法用到的公式如下：

$$Q(v)=\sum_{u\in\Gamma(v)}D(u) \tag{4-1}$$

$$I(v)=\alpha\ D(v)+\beta\quad Q(\tag{4-2}$$

$$S=\bigcup_{i=1}^{T}A(k_i) \tag{4-3}$$

$$\sum_{i=1}^{T}k_i=k \tag{4-4}$$

其中 k_i 表示给第 i 个社区分配的种子节点的个数，N_i 表示第 i 个社区节点的个数，N 表示整个网络中节点的个数。

对于式（4-1）中，$D(u)$表示节点 u 的度数，$Q(v)$表示节点 v 的邻居的度数总和。式（4-2）中 $I(v)$表示节点的重要度，α，β 表示权重因子，且 $\alpha+\beta=1$。式（4-3）中表示对各个社区上找到的影响力最大的种子节点集合 $A(k_i)$合并后得到整个网络上的种子节点集合 S。式（4-4）表示得到的种子节点集合 S 的节点个数，即初始种子节点个数。

4.3.3 算法流程图

本文提出影响力最大化算法，开始先输入社会网络数据集，然后根据本文所采用的社区划分算法，将整个社会网络划分为不同的社区结构，如果所划分的社区个数大于种子集合的节点个数 k 时，需要进行社区之间的合并，直到所划分的社区个数小于种子集合的节点个数 k 时，将所划分的社区结果输入到 MapReduce 模型中，再根据式（4-2）统计社区上每个节点的影响力值，最后将每个社区上得到的 $Top\text{-}k_i$ 种子节点集合合并，最终得到整个社会网络上的 $Top\text{-}k$ 种子节点集合。PC-NIE 算法的流程图如图 4.4 所示。

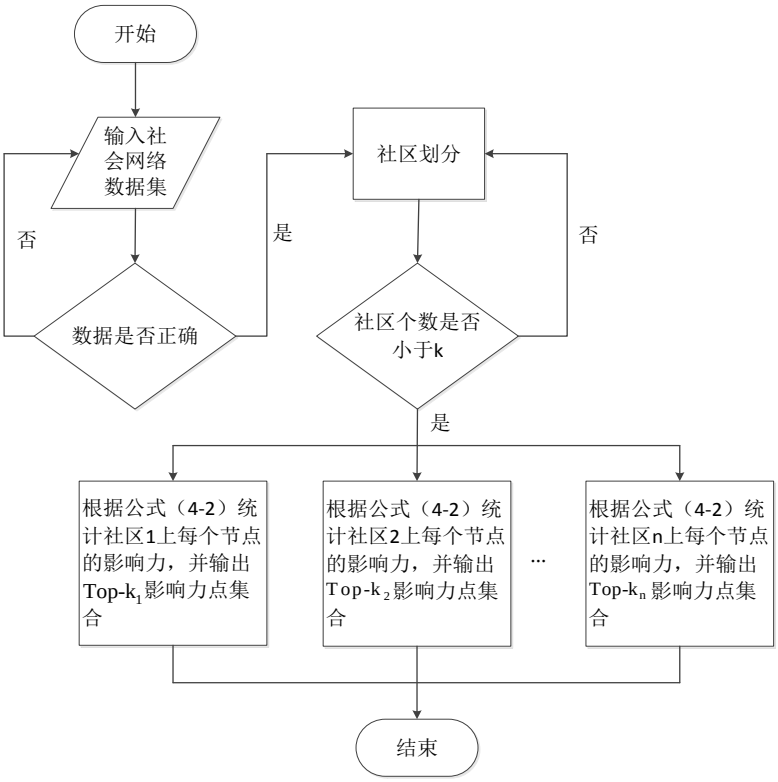

图 4. 4　影响力最大化算法流程图

4.4 算法伪代码描述

PC-NIE 算法的伪代码描述如算法 4.1 所示：

算法 4. 1　PC-NIE 算法

Algorithm4.1PC-NIE Algorithm
Input(G,k)
1:{C_i} =detect communities in G//使用 BGLL 算法进行社区划分
2: adjust{C_i}//调整社区
3: for i=1 to T do//将 k 按比例分配到各个社区上
4:　按比例给每个社区分配初始节点个数
5: end for
6:Parallel mining community C_i//并行挖掘每个社区上影响力最大的种子节点
7: for i=1 to T//将各个社区上得到的种子节点集合并
8:　$S \rightarrow S \cup A(k_i)$
9: end for
Output:S

第 1 行：采用快速发现社区算法将整个社会网络划分为不同的社区结构。

第 2 行：如果社区划分个数大于 k，则做社区的合并，直到社区个数小于 k。

第 3~5 行：将初始种子节点的个数根据社区的规模分配到各个社区上。

第 6 行：根据式（4-2）并行挖掘每个社区上影响力最大的前 k_i 个节点。

第 7~9 行：根据之前分配到的种子节点个数从各个社区上得到的影响力最大节点构成整个网络上的初始种子节点集合。

上述算法 4.1 中第 6 行的并行挖掘社区上影响力最大的节点集合。本文中所采用的并行编程模型为 MapReduce 模型，前面介绍过 MapReduce 是一个程序框架，该框架可以并行处理大规模的无结构数据，我们只需将数据保存在一个分布式系统中，在 HDFS 中，每个 MapReduce 任务都将被视为一个 Job。每个 Job 又被分为两个阶段：map 阶段和 reduce 阶段。Map 阶段和 reduce 阶段可以分别用 map 函数和 reduce 函数来表示。程序人员只需要设计好这两个函数即可实现数据的并行处理，这种编程模式实现起来比较容易。

为了得到节点的邻居节点集合和邻居的邻居集合，本文是通过两次 MapReduce 得到的，最后使用式（4-2）来计算节点的影响力还需要一次 MapReduce 得到 *Top-k* 节点集合，具体的算法如下。

表 4.1 中给出了统计节点邻居集合的描述。在任务的 Mapper 部分，输入形式为节点对<*u*,*v*>，文件中存放的数据是每一行表示图中的一条边，通过分割将节点对映射为<key，value>交给 Reducer 处理。

表 4. 1　获取节点邻居集合

获得一跳邻居节点集合	
Mapper	Input: u,v
	Key:line position Value:*u*,*v*
	output $<u,v>$
Reducer	Input: $<u,v>$
	Key: u Value: v
	output $<u,\partial(u)>$

在表 4.1 中的 Reducer 部分，reduce 的输入是 map 输出的键值对结果，reduce 函数的主要功能是将 key 值相同的 value 以空格分隔保存到一个字符串中输出，最后得到各节点的邻居集合。

为得到节点的两条信息，这里做一下转换，具体的转换格式为<节点，<节点的邻居-节点邻居的邻居>>，比如：节点 2 的邻居节点为 2，3，4 表示为<1，<2，3，4>> ，这是邻接表的一条记录，可以将它转换为以下形式：

<2，<1-3，4>>，<3，<1-2，4>>，<4，<1-2，3>>

具体的处理过程如表 4.2 所示：

表 4.2　获得两跳邻居节点度数之和

获得所有两跳邻居节点集合	
Mapper	Input:u,$\delta(u)$
	Key:line position Value: u,$\delta(u)$
	对获得的一跳邻居节点进行形式转换 output<v,u-{$\delta(u)$-v}>
Reducer	Input: <v,u-{$\delta(u)$-v}>
	Key: v Value: u-{$\delta(u)$-v}
	Reduce:output：节点的邻居节点度数之和

表 4.2 中给出了统计节点邻居集合的度数之和描述。在任务的 Mapper 部分主要是将节点的邻接表形式转换为<节点，<节点的邻居-节点邻居的邻居>>的形式，然后交给 Reducer 部分处理。在 Reducer 部分主要是将相同键值的节点合并，并统计节点邻居集合的度数总和。

至此，节点的度数和节点邻居的度数和都已处理完毕，还需要最后一次 MapReduce 来统计节点的影响力，并输出 *Top-k* 节点集合，具体的处理过程如表 4.3 所示。

表 4.3　得到 *Top-k* 节点集合

计算节点的影响力获取 *Top-k* 节点集合	
Mapper	Input：c_{id}-u，　$sum1$-$sum2$
	Key:line position Value: c_{id}-u，　$sum1$-$sum2$
	将 Value 分割为<Key，Value>形式 通过式（4-2）得到每个节点的影响力。 Output：c_{id}-u，　$sum1$-$sum2$
Reducer	Input：c_{id}-u，　$sum1$-$sum2$
	Key: c_{id}-u Value：$sum1$-$sum2$
	Reduce：根据 Key 排序得到各个社区上的 *Top-k* 节点集合。 Output：*Top-k* 节点集合

通过 MapReduce 模型，设计 map 和 reduce 函数计算节点的影响力，最终统计得到整个社会网络上的 *Top-k* 节点集合。

4.5 算法分析

算法的时间复杂度分析：假设算法 4.1 中第 1 行社区发现的时间复杂度为 $O(C)$，第

3~5 行时间复杂度为 $O(T)$，第 6 行的时间复杂度，根据表 4.1~表 4.3 分析，其时间复杂度为 $O(k_i n_i p_i)$，其中 p_i 表示节点邻居的个数，PC-NIE 算法的时间复杂度为 $O(C+T+k_i n_i p_i)$。

4.6 实验

在前面小节部分介绍了算法的思想描述，并给出了算法的流程图以及算法的伪代码描述。这一节将针对本文提出的算法搭建 Hadoop 实验环境，通过实验来验证本文提出算法的有效性。

4.6.1 Hadoop 实验环境搭建

本文的仿真实验采用的是 Hadoop 框架进行设计的，所搭建的集群环境中包括一个 Master 节点，多个 Slaver 节点。它们都布局在同一个局域网下，可以相互 ping 通，因此每台机器使用相同的用户名和密码即可，用户名都为 Administrator，密码为 123。Master 节点负责管理 DFS 的命名和客户端对 DFS 的访问与操作。Slaver 节点负责存储数据和计算。系统的架构如 4.5 所示：

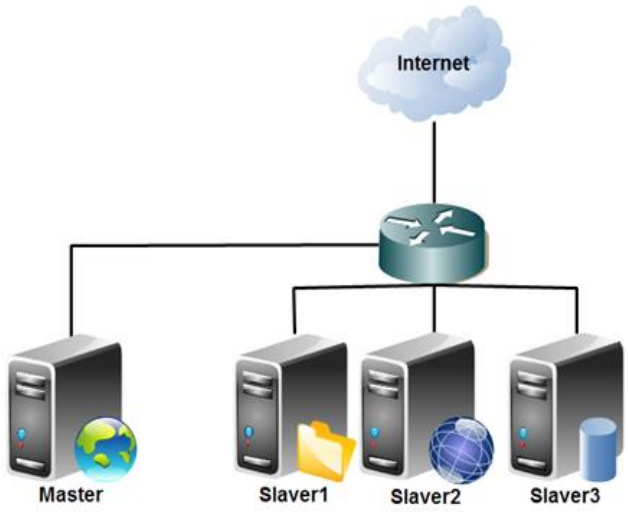

图 4. 5　集群环境架构图

1．节点的 IP 地址分布如表 4.4 所示：

表 4. 4　节点 IP 地址分布

机器名	IP 地址
Master	222.27.254.61
Slave1	222.27.254.152
Slave2	222.27.254.25
Slave13	222.27.254.145

2．集群环境搭建过程：

首先，在各台机器上安装如下软件：Windows7 操作系统，Cygwin，Hadoop-0.20.2 以及 JDK1.6 版本，并配置各软件的环境变量，如表 4.5 所示：

表 4. 5　环境变量配置

PATH	%JAVA_HOME%\bin; D:\cygwin\bin;D:\cygwin\usr\sbin;
JAVA_HOME	C:\Java\jdk1.6.0_10
CLASSPATH	.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\dt.jar; %JAVA_HOME%\lib\tools.jar

其次，配置 SSH 免密码登录。使用命令 ssh-keygen -t rsa 生成一个密钥对 id_rsa 和 id_rsa.pub，再通过命令 cp .ssh/id_rsa.pub .ssh/authorized_keys 把公钥追加到其他主机的 authorized_keys 文件中。将 id_rsa.pub 发送到集群其他节点计算机中，按照上述同样的操作方式得到 authorized_keys。这样集群中节点之间的免密码登录配置完成。

最后，hadoop 集群配置。当在各台机器上安装 Hadoop-0.20.2 之后，在 Hadoop-0.20.2 安装目录的 conf 目录下就可以配置 hadoop 文件了，主要配置的文件[52]有：hadoop-env.sh、core-site.xml、hdfs-site.xml、mapred-site.xml、masters 和 slaves。具体配置如下：

（1）修改 hadoop-env.sh，设置 Java 环境变量 exportJAVA_HOME=/home/Administrator /jdk1.6.0_10

（2）配置 masters 和 slaves 文件。在 masters 文件夹中添加 Master，在 slaves 文件夹中添加 Slave1、Slave2、Slave3。

（3）配置 core-site.xml、hdfs-site.xml 和 mapred-site.xml。

core-site.xml[51]：

```
<property>
<name>fs.default.name</name>
<value>hdfs:// Master:9000</value>
</property>
```

hdfs-site.xml：

```
<property>
 <name>dfs.replication</name>
  <value>4</value>
</property>
```

mapred-site.xml[53]：

```
<property>
```

```
 <name>mapred.job.tracker</name>
  <value> Master:9001</value>
</property>
<property>
 <name>mapred.tasktracker.map.tasks.maximum</name>
 <value>4</value>
</property>
<property>
 <name>mapred.tasktracker.reduce.tasks.maximum</name>
 <value>4</value>
</property>
```

至此，hadoop 集群配置就完成了，启动 hadoop 就可以运行 MapReduce 程序了，启动服务如图 4.6 所示：

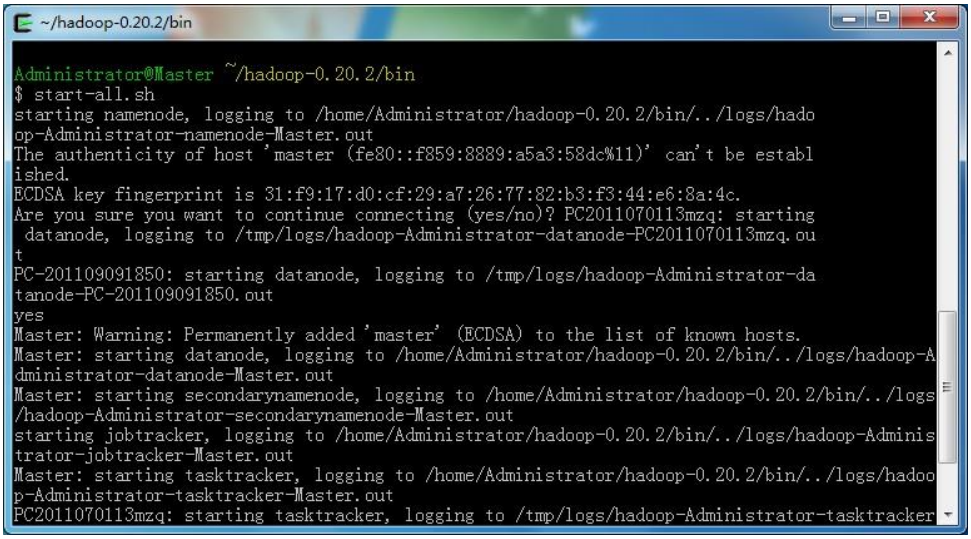

图 4.6　集群启动服务图

4.6.2 实验数据集

实验所采用的数据集是两个真实网络的数据集 Amazon product co-purchasing network 和 Wikipedia Talk network。这两个数据集都是从斯坦福大学的 SNAP 栏目下载获取的。

1．Amazon product co-purchasing network。该数据集是 2003 年 5 月 5 日亚马逊网上产品联合采购关系数据集，此网络是一个有向无权网络，网络中节点表示产品，产品之

间的边，表示两个产品经常在一起购买，网络的一些统计详情如表 4.6 所示。

表 4. 6　Amazon product co-purchasing network 各项指标统计详情

统计项	统计值
节点数	410236
边数	3356824
最大的 SCC 中的节点数（在所有节点中所占比例）	390304 (0.951)
最大的 SCC 中的边数（在所有边中所占比例）	3255816 (0.970)

2．Wikipedia Talk network。维基百科是一个由志愿者合作编写的免费的百科全书，每一个注册用户都有一个页面，他们中的任何人都可以编辑维基百科中的任何文章和条目，网络中包含了截止到 2008 年 1 月的所有用户编辑过的文章和条目，该网络中节点表示一个维基百科用户，从节点 i 指向节点 j 之间的边，表示用户 i 最近一次编辑过用户 j 的页面，网络中一些统计指标详情如表 4.7 所示。

表 4. 7　Wikipedia Talk network 各项指标统计详情

统计项	统计值
节点数	2394385
边数	5021410
最大的 SCC 中的节点数（在所有节点中所占比例）	111881 (0.047)
最大的 SCC 中的边数（在所有边中所占比例）	1477893 (0.294)

4.6.3 实验结果及分析

在这一节，将采用独立级联模型，并针对 Random 算法、DegreeDiscount（D-D）算法、CGA 算法以及本文提出的 PC-NIE 算法在真实的网络数据集上进行对比实验，并以图表的形式给出实验结果。

1．当种子集合 k 变化时的传播效果分析

下面两个实验将从传播效果上进行对比分析，种子节点个数 k 的变化范围是从 10 到 60 之间，传播概率设置为 0.05，参数 α 和 β 分别取值 0.5,0.5。

图 4.7 展示了 4 种算法在 Amazon 数据集上的运行结果，实验取 1000 次模拟传播的平均值。横坐标为种子节点选取个数 K，纵坐标为种子节点激活节点的数目。从图中可以看出，随机选取种子节点（Random）算法和其他 3 种算法比较，其传播效果最差，

这和在之前的工作中叙述的是一致的，DegreeDiscount（D-D）算法传播效果较优于 CGA 算法的传播效果，PC-NIE 算法的传播范围和其 3 中算法相比提高了大约 23%，从图 4.7 可以看出 PC-NIE 算法的效果明显优于其他 3 种算法，这是由于 CGA 算法利用社区划分算法将整个社会网络划分为独立的社区之后，将每个社区作为一个独立的网络来处理，这样就使得社区之间失去了联系，中断了社区之间的传播，而 PC-NIE 算法也是将整个网络划分为不同的社区，但是该算法保留了社区之间的联系，从而提高了网络传播的精度，因此 PC-NIE 算法的传播效果要比 CGA 算法的传播效果好。

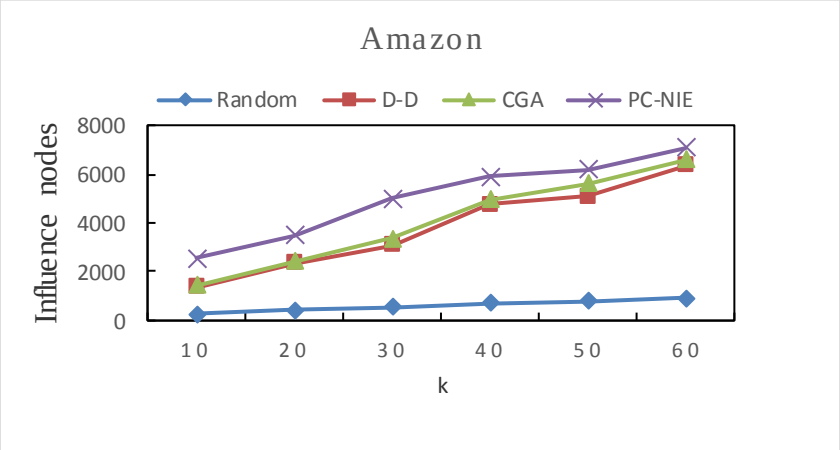

图 4. 7　不同算法在 Amazon 数据集上的运行结果

图 4.8 展示了 4 种算法在 Wikipedia 网络上的传播范围比较。实验同样取 1000 次模拟传播的平均值。横坐标为种子节点选取个数 K，纵坐标为种子节点激活节点的数目。同样从图中可以看出 PC-NIE 算法的传播效果要好于其他 3 中算法的传播效果，其传播范围提高了 15%。

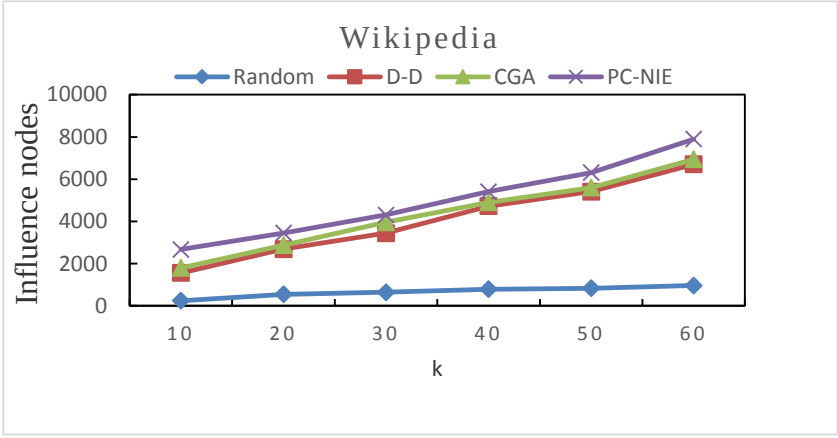

图 4. 8　不同算法在 Wikipedia 数据集上的运行结果

2．当传播概率 p 变化时的传播效果分析

图 4.9 是在 Amazon 数据集上，k=50 时，随着传播概率 p 的变化，4 种算法传播范

围的不同变化。随着 p 的变化，PC-NIE 算法和 CGA 算法与 Random 算法和 DegreeDiscount（D-D）算法的传播范围差距越来越大，当 p=0.09 时，CGA 算法的精度是 NewGreedy 算法的 78%，而本文的提出的 PC-NIE 算法的精度可达 NewGreedy 算法的 85%。因此本文提出的算法在精度上更接近于整个网络传播的 NewGreedy 算法。

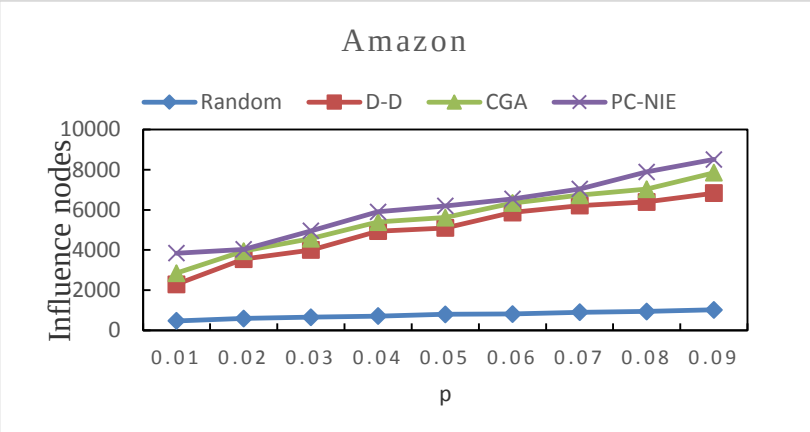

图 4. 9　随传播概率 p 变化的 4 种算法传播范围比较

图 4.10 是在 Wikipedia 数据集上，k=50 时，随着传播概率 p 的变化，4 种算法传播范围的不同变化。随着 p 的增长，Random 算法的传播范围没有明显增大，而本文提出的 PC-NIE 算法的传播范围与其他三种算法的差距越来越大，这是因为 CGA 算法将整个社会网络划分社区后，使得社区之间失去了联系，影响了整个网络的传播范围，而本文算法将网络进行社区划分后仍然保持社区之间的联系，加大了网络的传播范围。

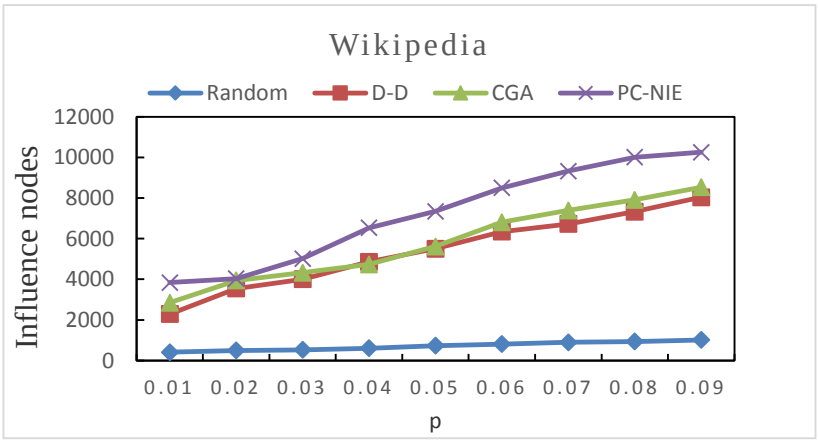

图 4. 10　随传播概率 p 变化的 4 种算法传播范围比较

3．算法执行时间分析

下面两个实验将从算法运行时间上进行对比分析，种子节点选取个数为 100 个，传播概率设置为 0.05。

不同算法在 Amazon 数据集上运行时间如图 4.11 所示：

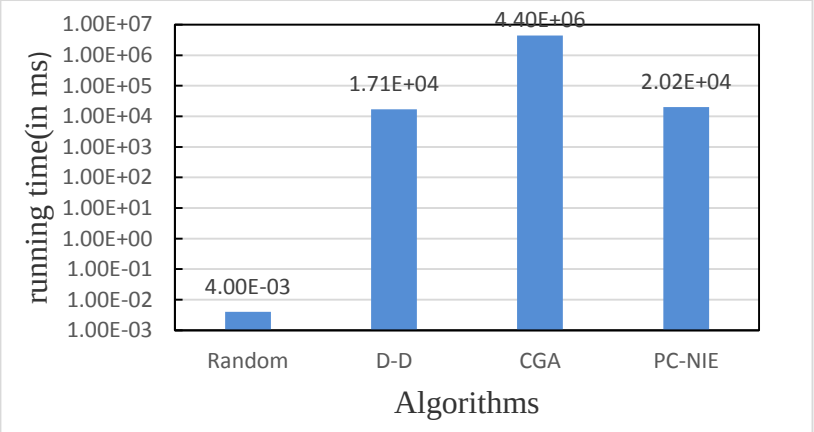

图 4. 11　不同算法在 Amazon 数据集上的运行时间

从图 4.11 可以看出 Random 算法的运行时间在这 4 种算法中运行时间最短，是因为该算法是随机选择种子节点，但是该算法和其他算法比较，其影响效果是最差的，其次是 DegreeDiscount（D-D）算法，虽然本文提出的算法运行时间比 DegreeDiscount（D-D）算法略低一些，但是和本文算法相比 DegreeDiscount（D-D）算法的影响效果要略显的差些，CGA 算法运行时间最慢，需要 1 个多小时。

图 4.12 是不同算法在 Wikipedia 数据集上运行时间结果。从图中可以看出 CGA 算法的运行时间是最长的，大约 1.4 个小时，这是由于 CGA 算法进行社区划分后，在各个社区上寻找影响力最大的种子节点时采用的贪心算法，因此时间消耗比较长，本文提出的算法远低于 CGA 算法，是由于本文提出算法将社区划分之后，并行实现在各个社区上寻找影响力最大的种子节点，并且在计算节点影响力时考虑的是节点的度数和节点邻居度数之和的加权。Random 算法运行效果依然是最高的，DegreeDiscount（D-D）算法次之。

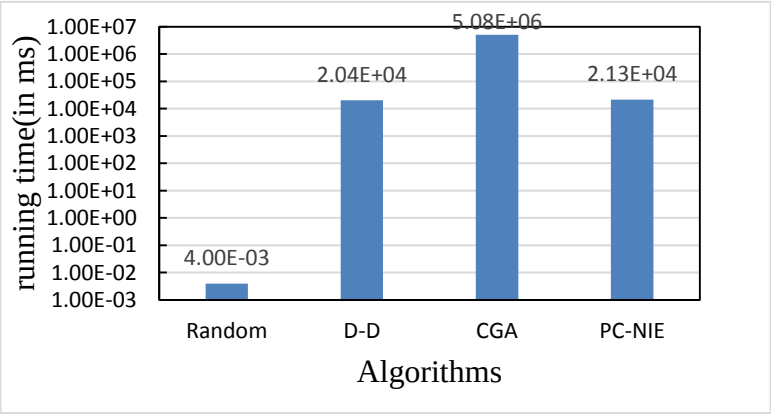

图 4. 12　不同算法在 Wikipedia 数据集上的运行时间

4.7 本章小结

本章基于 MapReduce 并行的特点，从并行的角度重新考虑了社会网络传播影响最大化问题，通过搭建 Hadoop 环境来进行实验，在真实的社会网络中实现挖掘 Top-k 节点的并行算法，实验结果表明本文提出的算法传播效果和运行时间两个方面都比 Random 算法、DegreeDiscount（D-D）算法以及 CGA 算法要好，这表明了本文提出的 PCINIE 算法比较好的解决目前影响力最大化算法中存在的不足，因此本文提出的算法可以用于解决信息传播最大化、微博营销等影响力最大化问题，具有一定的现实意义。

第5章 影响力最大化传播模型研究

信息的传播离不开特定的传播模型，本章将对影响力最大化传播模型展开研究，目前比较常用的传播模型是独立级联传播模型，但是独立级联模型中的影响概率 p 在整个传播过程中是不变的，然而现实生活中一个个体的影响力是随信誉的变化而发生变化。因此本章节将针对这个问题提出一种新的影响力传播模型：浮动传播级联模型。并在该模型上进一步研究社会网络中的影响力最大化问题。

5.1 存在的问题

独立级联模型一个比较经典的影响力最大化传播模型，模型传播过程为，在某一 t 时刻，如果节点 v 转变为活跃状态，则在下一个时刻 t+1，节点 v 尝试去影响它的不活跃邻居节点 u，成功影响的概率为 p_{vu}。我们将这一问题看成是以概率 p_{vu} 的偏差投掷硬币的问题，而对于社会网络中所有的边在什么时候“投掷硬币”最终传播结果不会产生影响，若将通路边看成是投掷成功对应的边，即节点 u 被成功影响，否则看作为阻路边，即节点 u 影响失败，如图 5.1 所示，

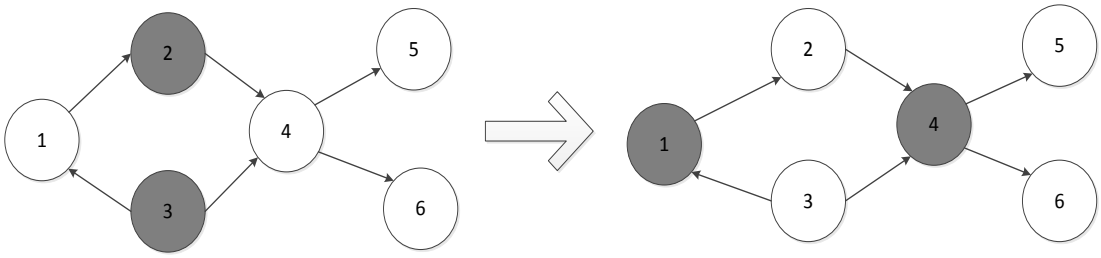

图 5. 1 **独立级联**模型信息传播图

上图中在初始时刻，有“影响力”的用户是用户 2 和 3，下一个时刻，用户 2 和 3 将以任意顺序向他们的邻居用户推荐新产品，成功推荐的概率是 p_{vu}，产品传播结束之后，用户 1 和 4 成功接受了来自用户 2 和 3 的推荐。然而独立级联模型中的影响概率 p_{vu} 是一个随机赋予的常数，它表示一个个体对其邻居的影响概率，它并不随着信息的传播而发生改变，在 FAP-IC 模型[54]中，影响概率 p_{vu} 是一个随着成功影响邻居个数的增加不断增加的。但是当一个公司对一个新产品进行宣传推广时，就会有这样的问题发生，一个个体的影响概率不但随着成功影响邻居数量的增加而增加，而且会随着尝试影响邻居失败人数的增加而降低。本文希望在传播过程中将尝试影响邻居失败人数也考虑进

来，这样就使得在信息在传播过程中，个体的影响力会随着影响成功而增强或者影响失败而减弱，使其更加符合传播过程的实际情况。

5.2 浮动传播级联模型（FTC）

5.2.1 FTC 模型思想

最近某公司生产了一批新产品，对于消费者是否愿意购买该产品，他们不但会考虑到产品推荐者过去的信誉，还会调查一下该产品目前的购买状况，购买该产品的人越多他越容易买，反之，他可能会放弃购买该产品。换句话说，影响概率会随着购买人数的涨幅而不断的更新，因此在本文对独立级联模型的改进是将活跃节点未能成功影响邻居节点的概率也考虑到影响概率的计算中。

FTC 模型的传播过程与 IC 模型和 FAP-IC 模型[55]的传播过程是基本一致的，在经典的独立级联模型中，影响概率 p_{vu} 是一个与传播过程无关的常量，然而，在改进的模型中影响概率 p_{vu} 的计算需要分为两步进行：第一步：在初始时刻给网络中的每一条边分配一个影响概率 p_{vu}；第二步：在每一个时间步 t 中，若活跃节点尝试影响它的任意一个不活跃邻居节点后，无论影响成功与否，在下一个时间步 t+1 时刻，该节点尝试影响它的另外一个不活跃邻居节点时，都要按照式（5-1）更新该邻居节点与该节点边上的影响概率 p_{vu}。

$$p'_{vu} = \left(1+\frac{k}{|N(v)|}\right)p_{vu} \tag{4-1}$$

式（5-1）中 p_{vu} 是上一时刻节点 v 对节点 u 的影响概率，$|N(v)|$是已成功影响节点的邻居节点个数，k 的取值为 1 或者-1，如果上一时刻节点 v 成功影响了其不活跃邻居节点，则 k 取值为 1，反之 k 取值为-1，p'_{vu} 是更新后的影响概率。

在上述节点影响概率的更新过程中，最终成功影响的节点数目与节点影响顺序有关，也就是说，节点影响具有顺序相关性，为了消除这一问题，本文采取的办法是多次实验取平均值。

5.2.2 FTC 模型举例

为了进一步理解 FTC 模型的思想，本节将给出一个例子来讲述 FTC 模型，如图 5.2，已知社会网络 $G(V,E)$，V={1,2,3,4,5,6,7,8}，在该网络中选选取 3 作为种子用户，下面给

出用户 3 向其 5 个邻居推荐产品的过程来理解 FTC 模型中影响概率 p_{32}、p_{34}、p_{35}、p_{36}、p_{37} 的变化情况。

在图 5.2 中，影响概率都是在传播的初始时刻随机赋值，其值分别为 p_{32}=0.15，$p_{34}=0.2$，p_{35}=0.3，p_{36}=0.2，p_{37}=0.4，在初始时刻，用户 3 随机选择一个邻居用户向他推荐产品。

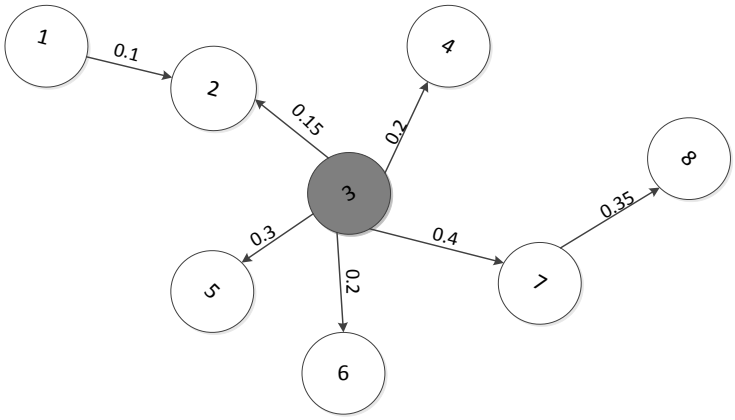

图 5. 2　社会网络初始状态

在图 5.3 中，用户 3 在初始时刻随机选择邻居用户 2，进行产品推荐，并且推荐失败，根据 FTC 模型，用户 2 未能接受来自用户 3 的推荐，根据式（5-1）更新用户 3 对其邻居的影响概率，p_{34}=(1-1/5)×0.2=0.16，p_{35}=0.24，p_{36}=0.16，p_{37}=0.32，此时用户 3 不再向用户 2 进行产品的推荐，即 p_{32}=0。

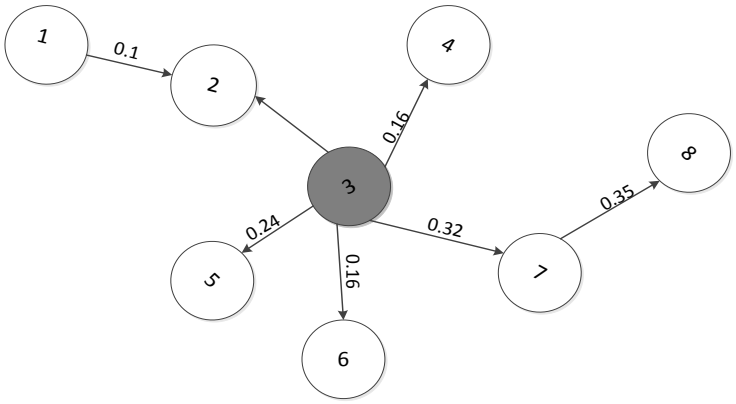

图 5. 3　用户 2 **拒绝**推荐产品后的概率更新图

在图 5.4 中，用户 3 随机选择邻居用户 7，向其进行产品推荐，推荐成功，根据式（5-1）更新用户 3 对其余邻居用户的影响概率，p_{34}=(1+1/5) ×0.16=0.192，p_{35}=0.288，p_{36}=0.192。

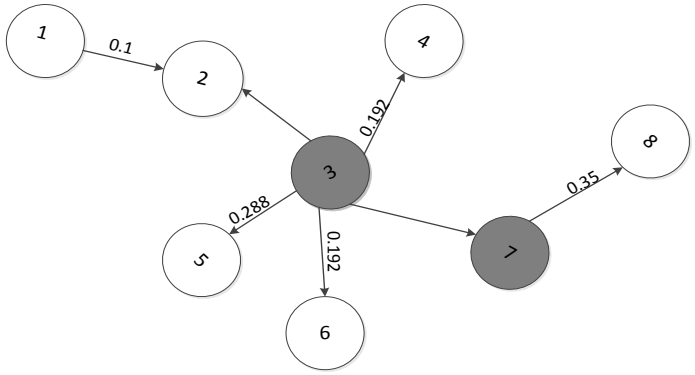

图 5. 4　用户 7 接受推荐产品后的概率更新图

在图 5.5 中，用户 3 随机选取邻居用户 5 进行产品推荐，并且推荐成功，根据式（4-1）更新用户 3 对其余邻居用户的影响概率，p_{34}=0.2304，p_{36}=0.2304

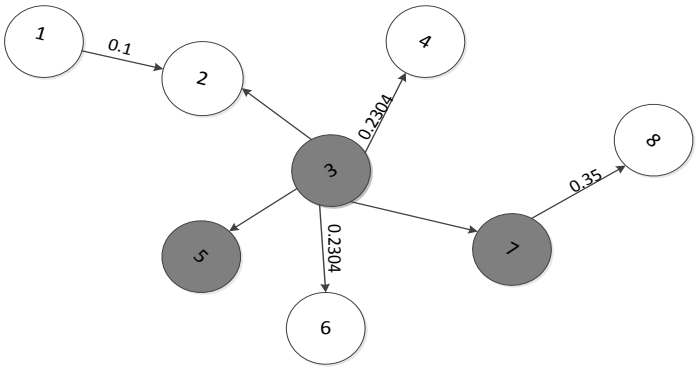

图 5. 5 用户 5 接受推荐产品后的概率更新图

在图 5.6 中，用户 3 从剩余邻居中随机选取用户 4 进行产品的推荐，并且推荐成功，根据式（5-1）更新用户 3 对其邻居用户 6 的影响概率，p_{36}=0.27648。

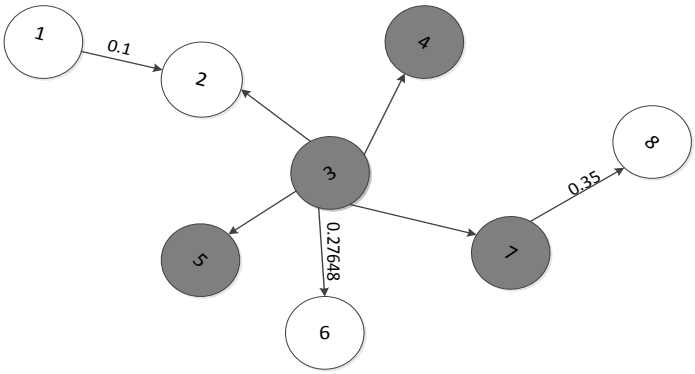

图 5. 6　用户 4 接受推荐产品后的概率更新图

在图 5.7 中，用户 3 对其邻居用户 6 进行产品推荐，推荐失败，此时用户 3 不再对用户 6 进行产品推荐，即 p_{36}=0，用户 3 对其邻居用户进行产品推荐的过程结束。

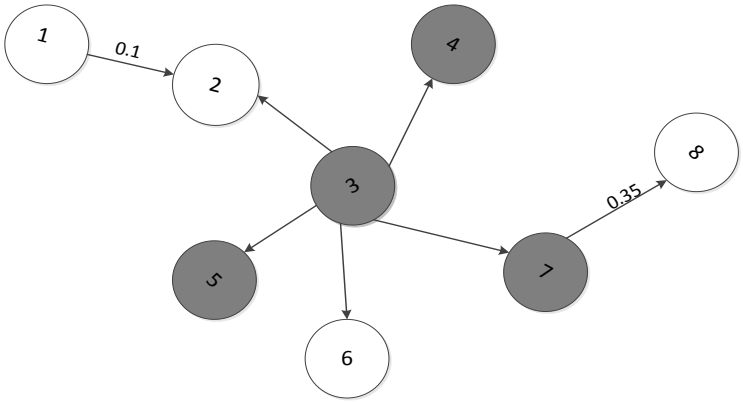

图 5. 7　用户 6 拒绝接受推荐后的概率更新图

用户 3 对其邻居进行产品推荐结束后，再从已接受该产品的用户中随机选取一个用户重复上述推荐过程，直到网络中不再有新的用户接受该产品，推荐过程结束。

5.3 实验

本节将 FTC 模型应用在真实的网络上，对 FTC 模型和 IC 模型上分别进行实验，最后给出实验结果并针对实验结果进行详细的对比分析。

5.3.1 实验方法

本文实验所用的软件工具和实验的运行环境如下：

硬件：Intel(R) Core(TM)2 CPU，2.67GHz 主频，2.0GB 内存，500G 硬盘

操作系统：windows7

软件：Eclipse4.3.0

本实验所有算法的代码实现都采用 Java 语言实现的。

5.3.2 实验数据集

本文实验采用了两个真实网络的数据集，这两个数据集是从斯坦福大学的 SNAP 栏目下载的，即 P2P-Gnutella08 数据集和 NEThep network 数据集。

1. Gnutella 数据集。这里 Gnutella 数据集并没有下载 Gnutella 网络上的全部数据集，而是截取的 2002 年 8 月的一部分数据。在该数据集中节点表示 Gnutella 中的主机，节点之间的边表示 Gnutella 中两台主机的连接关系。Gnutella 网络是一个无向无权网络，该网络的一些统计详情如表 5.1 所示。

表 5. 1　Gnutella 数据集的各项统计详情

统计项	统计值
节点数	6301
边数	20777
最大的 SCC 中的节点数（在所有节点中所占比例）	2068（0.328）
最大的 SCC 中的边数（在所有边中所占比例）	9313（0.448）

2．NETHep network 一个高能物理学家之间的学术论文合作网络，该网络中每个节点代表一个作者。如果作者 i 和作者 j 之间合作了一篇论文，那么在节点 i 和节点 j 就会形成一条无向边。如果 k 个作者合作一篇论文，那么就产生一个由 k 个节点组成的完全子图。该数据集包含了 1993 年 1 月至 2003 年 4 月（共 124 个月）的论文合作信息。具体的一些统计详情如表 5.2 所示：

表 5. 2　NETHep network 数据集各项指标统计详情

统计项	统计值
节点数	12008
边数	237010
最大的 SCC 中的节点数（在所有节点中所占比例）	11204（0.933）
最大的 SCC 中的边数（在所有边中所占比例）	117649（0.933）

5.3.3 实验结果及分析

作为对比实验，本文对比较经典的独立级联模型以及本文改进的模型，即浮动传播级联模型在 Gnutella 数据集和 NETHep network 数据集上分别进行实验，并以图表的形式给出实验结果，从模型效果上进行了对比实验。实验中种子集合的选取采用的是本文在第 3 章提出的算法。

图 5.8 所示，假定影响概率 p=0.05 不变，随种子节点个数 k 不断增长，FTC 模型和 IC 模型在 Gnutella 数据集上成功影响节点数目的对比图。

将影响概率 p 设为固定，在实验结果可以看出，随着种子节点个数 k 的增长 FTC 模型和 IC 模型成功影响节点的数目均呈上升趋势，但是本文提出的模型明显要优于独立级联模型，这是因为本文提出的模型在每次的影响过程结束，无论影响成功与否都将更新节点的影响概率，使得影响过程更符合实际的社会网络传播过程。

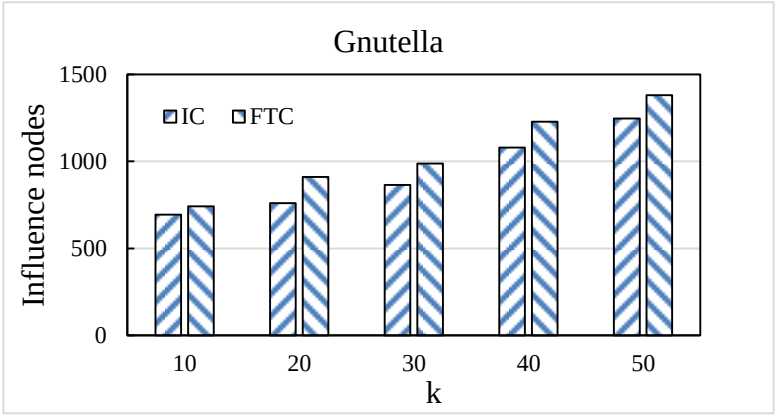

图 5. 8　Gnutella 数据集上 p 值固定的影响力传播结果

图 5.9 所示是在设定影响概率 p=0.05 不变的情况下，随种子集合节点个数 k 不断增加时 2 种传播模型在 NETHep network 数据集上成功影响节点数目的对比图。

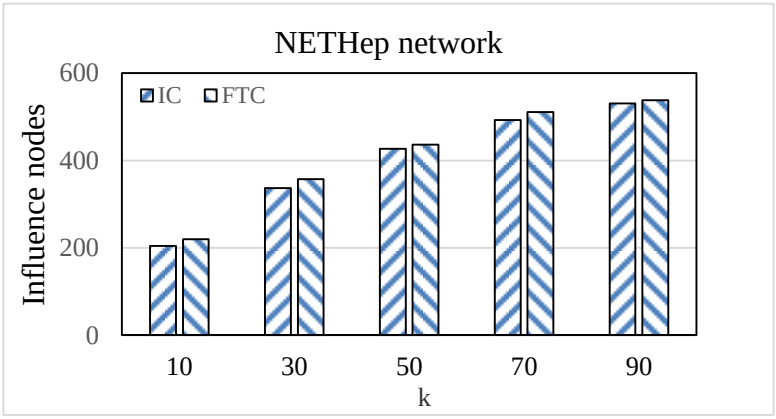

图 5. 9　NETHep network 数据集上 p 值固定的影响力传播结果

图 5.9 所示，在实验中，将影响概率 p 设为不变，随着种子集合的节点个数 k 的增长 2 种模型成功影响节点的数目均呈上升趋势，在相同的种子节点个数情况下，本文提出的 FTC 模型成功影响节点的数目要大致等同于 IC 模型成功影响节点的数目，这说明本文提出的模型在社会网络影响力最大化问题上具有有效性和可行性。

图 5.10 展示了在设定种子节点集合个数 k=30 不变的情况下，随着节点影响概率不断增加的情况下，两种传播模型在 Gnutella 数据集上成功影响节点数目的对比图。

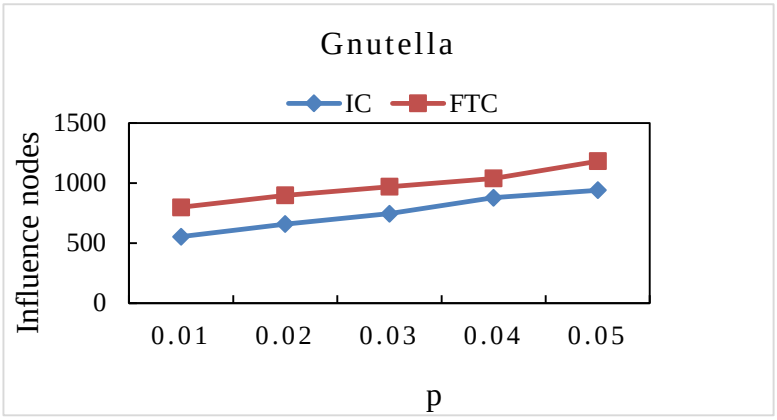

图 5. 10　Gnutella 数据集 *k* 值固定的影响力传播结果

图 5.10 所示，在实验中，将种子节点集合的节点个数 *k* 设为固定不变的值，随着影响概率 *p* 的增大，本文提出的 FTC 模型成功影响节点的数目明显要多于 IC 模型，从图中可以看出，提高节点的影响概率有利于社会网络中信息的传播。

图 5.11 展示了在设定种子节点集合个数 *k*=30 不变的情况下，随着节点影响概率不断增加的情况下，两种传播模型在 NETHep network 数据集上成功影响节点数目的对比图。

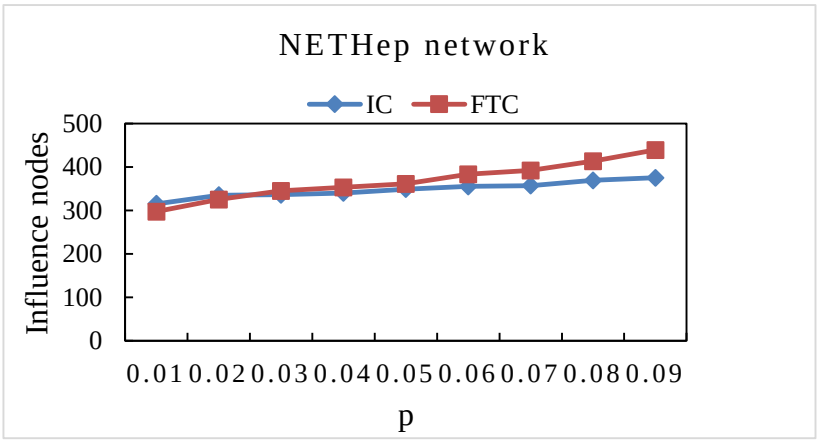

图 5. 11　NETHep network 数据集 *k* 值固定的影响力传播结果

图 5.11 所示，在实验中，将种子节点集合的节点个数 *k* 设为固定不变的值时，随着影响概率 *p* 的增大，种子集合成功影响节点的数目的变化结果，从图中可以看出当影响概率 $p<0.03$ 时，独立级联模型成功影响节点的数目要高于浮动传播级联模型，当影响概率 $p\geq0.03$ 时，浮动传播级联模型成功影响节点的数目要高于独立级联模型，从整体来看，本文提出的浮动传播级联模型具有一定的有效性和可行性。

5.4 本章小结

传播模型描述的是信息在社会网络中的传播过程，主要应用于微博网络中的产品营销或者病毒营销中产品的推荐等。本章主要针对现有模型存在的不足，提出了一种浮动传播级联模型（FTC），并利用该模型进行社会网络影响力最大化研究，该模型的主要思想是在传播过程中，根据活跃节点在上一时刻影响不活跃邻居节点成功和失败的情况不断的更新模型中的影响概率，使得该模型更加符合真实网络中的传播规律。最后在两个真实的网络数据集上对本文改进的模型和 IC 模型做了对比实验，验证了本文改进的传播模型在社会网络影响力最大化问题上具有一定的可行性和有效性。

结论

本文深入研究了社会网络影响力最大化算法及影响最大化传播模型，在此基础上提出一种影响力最大化算法和一种影响最大化传播模型，并根据实验中的不足，提出进一步的研究方向。

本文从社会网络领域的影响力最大化问题出发，详细描述了两个比较经典的社会网络传播模型及近些年主要的影响力最大化的相关算法，根据现有算法存在的问题提出了一种影响力最大化算法及一种改进的影响力最大化传播模型：并行社区搜索影响力最大化算法和浮动传播级联模型。

并行社区搜索影响力最大化算法首先将社会网络划分为不同的社区，然后在各个社区上使用节点的度和节点邻居的度数和来选择影响力的节点，这种选择是并行实现的。在这个算法中社区划分算法使用的是目前最快的社区划分算法，BGLL 算法，当社区划分的个数大于选择的影响力节点个数 k 时，就将小社区进行合并，并最终使得社区划分的个数小于 k，然后将 k 个影响力节点按照社区的规模均匀的分配到各个社区上，最后将在各个社区上搜索到的影响力节点合并得到要选择的 k 个影响力节点。通过两个真实网络的数据集进行对比实验，证明了该算法不但保证了计算结果还提高了算法的运行效率，使得算法具有有效性和可行性。

浮动传播级联模型综合考虑了活跃节点推荐成功与推荐失败的情况，也就是说活跃节点对不活跃节点的影响概率是随着活跃节点影响成功和失败的次数而不断上下浮动的，这种考虑使得传播过程更符合实际的传播规律，是对经典独立级联模型的完善和补充。通过在真实的网络数据集上进行对比实验，证明了该模型具有一定的可行性和有效性。

本文主要研究了社会网络影响力最大化问题和社会网络影响力传播模型，这两个方面近些年在社会网络领域已有不少学者研究，他们提出了不同的影响力最大化算法和不同的影响力传播模型，本文在之前学者研究的基础上提出了一种并行社区搜索影响力最大化算法和一种浮动传播级联模型，虽然本文提出的算法通过对比实验在一定程度上具有可行性和有效性，但是由于笔者学术水平有限，本文还存在着很多不足之处，现将本文存在的问题总结如下：

1．在并行社区搜索影响力最大化算法中，本文首先采用现有的社区划分算法将整

个社会网络进行社区划分，然后将划分结果输入到 MapReduce 模型中，这就使得该算法不能完全在 Hadoop 环境中实现，将社区划分也纳入到 MapReduce 模型中实现完全并行算法是本文下一步研究的工作。

2．在浮动传播级联模型中，每次影响概率的更新都要根据节点上一时刻影响成功或者失败行为来更新，这就会使得传播结果与节点被影响的顺序相关，在今后的工作中，还需要改进该影响概率，使节点受影响的次序不影响最终的传播结果。

3．目前在社会网络领域研究的影响力最大化算法大都是针对静态社会网络进行研究的，然而真实网络都是随时间动态变化的，因此本文今后将针对一些动态社会网络进行研究社会网络影响力最大化问题和影响力传播模型。

参考文献

[1] Domingos P,Richardson M.Mining the network value of customers.Proceedings of the seventh Acm Sigkdd International Conference on Knowledge Discovery and Data Mining,San Francisco,Ca,USA,2001.New York,NY,USA:ACM:57–66P

[2] Kempe D,Kleinberg J,Tardos É.Maximizing the spread of influence through a social network.Proceedings of the ninth Acm Sigkdd International Conference on Knowledge Discovery and Data Mining.Washington,DC,USA,2003.New York,NY,USA:ACM: 137–146P

[3] Leskovec J,Krause A,Guestrin C.Cost-effective outbreak detection in networks. Proceedings of the 13th Acm Sigkdd International Conference on Knowledge Discovery and Data Mining.San Jose,CA,USA,2007.New York,NY,USA:ACM:420-429P

[4] Chen W,Wang Y,Yang S.Efficient influence maximization in social networks. Proceedings of the 15th Acm Sigkdd International Conference on Knowledge Discovery and Data Mining.Paris,France,2009.New York,NY,USA:ACM: 199–208P

[5] Kimura M,Saito K.Tractable models for information diffusion in social networks. Knowledge Discovery in Databases. 2006, 4213: 259–271P

[6] Narayanam R, Narahari Y. A shapley value-based approach to discover influential nodes in social networks.IEEE Transactions on Automation Science and Engineering. 2010,(99): 1–18P

[7] 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.计算机学报.2011,34(10): 1956–1965 页

[8] 陈浩,王轶彤.基于阈值的社交网络影响力最大化算法.计算机研究与发展.2012,49(10):2181-2188 页

[9] Newman M E J,Girvan M.Finding and evaluating community structure in networks. Physical review E.2004, 69(2): 026113-026128P

[10] Scripps J,Tan P-N,Esfahanian A-H.Node roles and community structure in networks. Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis.San Jose,CA,USA.2007.New York,NY,USA:ACM:26-35P

[11] Galstyan A,Musoyan V,Cohen P.Maximizing influence propagation in networks with community structure. Physical Review E.2009, 79(5): 056102-056109P

[12] Cao T,Wu X,Wang S.OASNET:an optimal allocation approach to influence maximization in modular social networks.Proceedings of the 2010 ACM Symposium on Applied Computing. Sierre,Switzerland.2010.New York,NY,USA:ACM:1088-1094P

[13] Wang Y,Cong G,Song G.Community-based greedy algorithm for mining top-k influential nodes in mobile social networks.Proceedings of the 16th Acm Sigkdd International Conference on Knowledge Discovery and Data Mining.Washington DC,DC,USA.2010.New York,NY,USA:ACM:1039–1048P

[14] 周夏冰,宋国杰,谢昆青.面向大规模社会网络 Top-k 影响力节点挖掘的并行算法.第 29 届中国数据库学术会议论文集（B 辑）,中国合肥,2012.北京：计算机研究与发展:215-221 页

[15] 汪小帆,刘亚冰.复杂网络中的社团结构算法综述.电子科技大学学报.2009,38(5): 537–543 页

[16] Blondel V D,Guillaume J-L,Lambiotte R.Fast unfolding of communities in large networks.Journal of Statistical Mechanics: Theory and Experiment.2008,2008(10): 10008-10020P

[17] 汪小帆,李翔,陈关荣.复杂网络理论及其应用.北京:清华大学出版社, 2006:4-5 页

[18] Watts D J 著,陈禹译.六度分隔:一个相互连接的时代的科学.北京:中国人民大学出版社, 2011:88-95 页

[19] 杨波.复杂社会网络的结构测度与模型研究.上海交通大学学位论文.2007:18-20 页

[20] 约翰·斯科特著,刘军译.社会网络分析法.重庆大学出版社,2007:68-81 页

[21] 罗家德.社会网络分析讲义.北京:社会科学文献出版社,2005:187-198 页

[22] Newman M E.The structure and function of complex networks.SIAM review.2003, 45(2):167–256P

[23] Newman M E.The structure of scientific collaboration networks.Proceedings of the National Academy of Sciences.2001, 98(2):404–409P

[24] Watts D J,Strogatz S H.Collective dynamics of “small-world”networks.Nature.1998, 393(6684): 440–442P

[25] Barabâsi A-L,Jeong H,Néda Z.Evolution of the social network of scientific

collaborations.Physica A:Statistical Mechanics and its Applications.2002,311(3):590-614P

[26] Ebel H,Mielsch L-I,Bornholdt S.Scale-free topology of e-mail networks.Physical Review E.2002,66(3):035103-035107P

[27] 杨建梅,陆履平,谢王丹.广州软件企业竞争关系的复杂网络分析.第二届全国复杂动态网络学术论坛论文集,中国北京,2005.中国学术期刊电子出版社: 616–618 页

[28] Parhi M.Dynamics of inter-firm linkages in Indian auto component industry: a social network analysis.DRUID Winter Conference.Aalborg East Denmark.2005.DRUID: 27–29P

[29] Fortunato S,Barthelemy M.Resolution limit in community detection.Proceedings of the National Academy of Sciences.2007,104(1):36–41P

[30] Newman M E.Analysis of weighted networks.Physical Review E.2004,70(5): 056131-056140P

[31] Leicht E A,Newman M E J.Community structure in directed networks.Physical Review Letters.2008,100(11):118703-118707P

[32] Kaplan T D,Forrest S.A dual assortative measure of community structure.arXiv preprint arXiv.2008,08(01).3290-3297P

[33] Newman M E J.Detecting community structure in networks.The European Physical Journal B-Condensed Matter and Complex Systems.2004,38(2):321–330P

[34] Sales-Pardo M,Guimera R,Moreira A A.Extracting the hierarchical organization of complex systems.Proceedings of the National Academy of Sciences.2007,104(39): 15224–15229P

[35] Richardson M,Domingos P.Mining knowledge-sharing sites for viral marketing[A]. Proceedings of the eighth Acm Sigkdd International Conference on Knowledge Discovery and Data Mining.Edmonton,AB,Canada.2002.New York,NY,USA:ACM: 61–70P

[36] Kempe D,Kleinberg J,Tardos É.Influential nodes in a diffusion model for social networks.Automata, Languages and Programming.2005,3580:1127-1138P

[37] Watts D J.A simple model of global cascades on random networks.Proceedings of the National Academy of Sciences.2002,99(9):5766–5771P

[38] Granovetter M.Threshold models of collective behavior.American journal of sociology.

1978,83(6):1420–1443P

[39] 冀进朝,韩笑,王喆.基于完全级联传播模型的社区影响最大化.吉林大学学报:理学版.2009,47(005):1032–1034 页

[40] Chen W,Collins A,Cummings R.Influence maximization in social networks when negative opinions may emerge and propagate.SIAM International Conf.on Data minging.Mesa (AZ),USA.2011:68-75P

[41] Even-Dar E,Shapira A.Internet and Network Economics.Springer,2007:281–286P

[42] Carnes T,Nagarajan C,Wild S M.Maximizing influence in a competitive social network: a follower's perspective.Proceedings of the ninth international conference on Electronic commerce.Minnerapolis,MN,USA.2007.New York,NY,USA:ACM:351-360P

[43] Nemhauser G L,Wolsey L A,Fisher M L.An analysis of approximations for maximizing submodular set functions—I.Mathematical Programming.1978,14(1): 265–294P

[44] Woeginger G J.Combinatorial Optimization—Eureka, You Shrink!.Springer,2003: 185–207P

[45] Lin N.Foundations of social research.McGraw-Hill New York,1976:34-40P

[46] Stonedahl F,Rand W,Wilensky U.Evolving viral marketing strategies.Proceedings of the 12th annual conference on Genetic and evolutionary computation.Portland,USA.2010. New York,NY,USA:ACM:1195–1202P

[47] Goyal A,Lu W,Lakshmanan L V.Simpath:An efficient algorithm for influence maximization under the linear threshold model.Data Mining (ICDM),2011 IEEE 11th International Conference on.Vancouver,Canada.2011.USA.IEEE Xplore:211–220P

[48] Ruan J,Zhang W.An efficient spectral algorithm for network community discovery and its applications to biological and social networks.Data Mining,2007.ICDM 2007.Seventh IEEE International Conference on.Omaha NE,USA.2007.USA.IEEE Xplore:643–648P

[49] 董西城.hadoop 技术内幕深入解析 MapReduce 架构设计与实现原理.机械工业出版社, 2013:31-38 页

[50] 赖海明.MapReduce 作业调度算法分析与优化研究.杭州电子科技大学学位论文.2013:1-2 页

[51] 张明辉.基于 Hadoop 的数据挖掘算法的分析与研究.昆明理工大学学位论文.2012:7-8 页

[52] 刘佳旭,刘万军.基于云计算平台的学生作业提交系统设计与实现.微计算机信息.2010,26(9-3):147–149 页

[53] 陈耀兵,刘斌,史延涛.基于 Hadoop 架构的大数据量日志存储和检索优化.信息网络安全.2013,6(10):40–45 页

[54] 马寅.社会网络影响力最大化算法及传播模型的研究.兰州大学学位论文.2012:33-36 页

攻读硕士学位期间发表的论文和取得的科研成果

致谢

时光荏苒如白驹过隙，转眼间我的研究生生活即将结束，回想第一次来到哈尔滨工程大学，我被这里的校园美景深深的吸引；回想起第一次走进软件支持研究室情景，仿佛就在昨天，在这里两年多的时间里，不但提高了我做项目的实践能力而且还提高了我做学术的学术能力；回想在这里生活的点点滴滴，它构成了我的整个研究生生活，增加了我的知识，丰富了我的人生。

在毕业论文即将完成之际，在此我要向我的导师董宇欣副教授表达我深深的谢意。董老师平日里工作繁忙，但她仍然从百忙之中抽出时间给予我们学术上和生活上无微不至的关怀，每周的学术研讨董老师都会去参加，她都是在认真听我们讲完之后给予我们关键性的指导意见和建议，在董老师细心的指导下，我们研讨小组的每个成员在学术上都得到了迅速的成长。董老师为人谦和、平易近人的形象令我倍感亲切和尊敬，我珍惜这份师生友谊，在这里我衷心的祝愿老师您工作顺利，健康幸福。

感谢我的父母，如果没有你们这些年对我的默默支持就没有现在的我，在今后的日子里我将继续努力。女儿希望你们健康快乐。

感谢实验室的韩启龙老师，在和韩老师一起开发项目的那段时间里，感谢他在技术上给予我们细心的指导和他一直以来在生活上给予的关怀。

感谢王莹洁师姐、张亚楠师兄、张伟伟同学和社会网络研究小组的师弟师妹们以及在学习上和生活上给予我帮助和支持的其他同学们，因为有你们，才使我的研究生生活那么的充实和快乐。

最后，衷心感谢在百忙之中审阅论文的老师，谢谢！