

# 黑龙江大学

## 本科毕业生毕业论文

论文题目： 社交网络结构洞关键技术的研究与应用

---

学 院： 计算机科学技术学院

---

年 级： 2013 级

---

专 业： 计算机科学与技术

---

姓 名： 王雪

---

学 号： 20132573

---

指导教师： 谭龙

---

2017 年 5 月 12 日

## 摘要

结构洞理论概念认为：一个个体属于连接群体与群体的桥梁，如果没有这个个体，这些群体将不再连通，那么这个个体可以从这种结构中受益。从网络整体来看，就好像网络结构中出现了一个洞穴。处于结构洞位置的个体将会从中受益，他们掌握了许多方面的最新消息，因此找到结构洞个体是非常有意义的。

结构洞对网络结构的分析有着非常重要的作用，它推动了社会网络以及社会资本理论的发展。综合国内外的发展现状，本文介绍了社交网络中的结构洞理论以及其算法，并结合 HIS 算法以及 BICC 算法提出了 HIS\_BICC 算法，利用 Python 来实现算法，使用 Twitter 数据集对各个算法从 precision, recall, F1\_score 以及伯特的结构洞衡量指标方面进行了性能分析。进而将结构洞理论应用于互联网产品，提出其在互联网产品推广中能够解决的问题以及其应用前景。结构洞在中国的应用大多数集中在情报分析领域，即网络安全方面，本论文试着从全新的方面对结构洞理论进行应用，运用 Flask 框架以及 D3js 库构建了一个基于结构洞的互联网产品应用。我们日后将验证结构洞理论在互联网产品运作中的作用。

## 关键词

结构洞；社交网络；社团结构；逆亲密中心；Python

# **Abstract**

The concept of the structural hole theory holds that an individual is a bridge between the group and the group, without which the group would not be connected. It seems like a cave in the network structure from the whole network. The individual in the hole of the structure will benefit from it, and they have mastered the latest news in many aspects, so it is very meaningful to find the structural holes.

Structural holes have a very important role in the analysis of network structures, It promotes the development of social networks and social capital theory. Integrated domestic and international development status, This paper introduces the structure of the social network theory and structural hole algorithm, and use Python to achieve the algorithm ,then we proposed HIS\_BICC algorithm on the basis of HIS algorithm and BICC algorithm, then we calculate the precision, recall, F1\_score and Burt's structural hole measure aspects of each algorithm for the performance analysis with the data set of Twitter. And then apply the theory of structural hole to the Internet products, we put forward its problems in the promotion of Internet products and its application prospects. Most of the applications of structural holes in China are concentrated in the field of information analysis, that is, network security. This paper tries to apply the theory of structural holes from a new aspect, and we use Flask framework and D3js Library constructed an Internet application based on structural holes. We will verify the structural hole theory in the operation of Internet products.

# **Key words**

structural hole; social network; community structure; inverse closeness centrality; Python

# 目录

|                        |    |
|------------------------|----|
| 摘要 .....               | I  |
| Abstract .....         | II |
| <br>                   |    |
| 第一章 绪论 .....           | 1  |
| 1.1 研究目的和意义 .....      | 1  |
| 1.2 国内外研究现状及发展动态 ..... | 1  |
| 1.3 主要研究内容 .....       | 2  |
| 1.4 本文组织结构 .....       | 2  |
| 第二章 相关研究理论基础 .....     | 3  |
| 2.1 社交网络 .....         | 3  |
| 2.2 结构洞 .....          | 3  |
| 2.3 PageRank 算法 .....  | 4  |
| 2.4 DFS 算法 .....       | 6  |
| 2.5 本章小结 .....         | 6  |
| 第三章 结构洞算法 .....        | 7  |
| 3.1 HIS 算法 .....       | 7  |
| 3.1.1 基于社团的问题定义 .....  | 7  |
| 3.1.2 HIS 算法思想 .....   | 8  |
| 3.1.3 HIS 算法分析 .....   | 9  |
| 3.2 ICC 算法 .....       | 11 |
| 3.2.1 基于图结构的问题定义 ..... | 11 |
| 3.2.2 ICC 算法思想 .....   | 13 |
| 3.2.3 ICC 算法分析 .....   | 14 |
| 3.3 BICC 算法 .....      | 15 |
| 3.3.1 BICC 算法思想 .....  | 15 |
| 3.3.2 BICC 算法分析 .....  | 16 |
| 3.4 本章小结 .....         | 17 |

第四章 结合 HIS 以及 BICC 的创新 HIS\_BICC 算法..... 18

4.1 HIS\_BICC 算法 ..... 18

4.1.1 HIS\_BICC 算法思想 ..... 18

4.1.2 HIS\_BICC 算法分析 ..... 19

4.2 实验环境搭建与测试 ..... 20

4.2.1 测试环境描述 ..... 20

4.2.2 实验测试数据 ..... 20

4.3 算法性能对比分析 ..... 20

4.3.1 算法运行时间比较 ..... 21

4.3.2 算法查准率比较 ..... 22

4.3.3 算法查全率比较 ..... 23

4.3.4 算法 F1\_score 比较 ..... 23

4.3.5 伯特的结构洞指标 ..... 24

4.3.6 算法性能综合分析 ..... 25

4.4 本章小结 ..... 26

第五章 结构洞在互联网产品上的应用 ..... 27

5.1 应用分析 ..... 28

5.2 预期结果 ..... 32

结论 ..... 33

参考文献 ..... 34

致谢 ..... 36

# 第一章 绪论

## 1.1 研究目的和意义

随着网络及各种技术的蓬勃发展，社交网络迅速的抢占了人们的视野，并得到了广泛的应用，如何利用社交网络显得尤为重要。对于现实生活中的企业家来说，我们假设他头脑灵活，那么他一般都会占据没有直接关系的两个个体之间位置，就好像我们现在所说的中介担当着中间人的角色，从而占据了“结构洞”位置。当他占据的结构洞越多，也就是充当了越多人的中介，他在周边的人际关系中就占据了不可替代的位置，调动周边人际关系的能力也就越强。一些实验研究已经验证，结构洞在信息传播中扮演了重要的角色。因此研究一个给定的社会网络中的结构洞的检测是非常有意义的。它可以应用在很多方面，如社区内核检测，传播控制，网络安全，和病毒式营销等，如果对其中的结构洞节点加以管理控制，可以有效的达到想要的目标。

## 1.2 国内外研究现状及发展动态

挖掘社会网中的结构洞的研究并不是很多，目前对于结构洞理论的算法设计可分为下面几类：最初 T. Lou 和 J. Tang<sup>[1]</sup>基于社团的理念提出挖掘社会网络中的结构洞算法 HIS。HIS 挖掘结构洞算法根据两步信息流理论：意见领袖首先会先接收到最新消息，然后消息再由意见领袖传播到其他的普通用户；从直觉上来讲，由于结构洞在不同社团之间的信息扩散起着重要作用，所以提出了 MaxD 算法。根据度中心性理论 Albert<sup>[2]</sup>认为选择 k 个度最大的节点就是 top-k 个结构洞。Mojtaba Rezvani<sup>[3]</sup>等人根据节点到其它所有节点的平均距离寻找结构洞，提出 ICC 算法，结合 L 界限长的邻居(即 L 步长能达到的邻居)来改善算法的性能，提出了 BICC 算法，最后结合网络中的关节点，提出了 AP\_BICC 算法。它们计算网络中所有节点对之间的平均距离，删除某个顶点时会使网络的平均距离增加最大化，那么删除的这个节点可以当作结构洞节点。L.Page 等<sup>[4]</sup>提出 PageRank 算法，使用节点的 PageRank 值来挖掘结构洞，认为节点的 PageRank 值就是节点被访问的可能性大小，选出 PageRank 值前 k 大的点。S.Goyal, F.Vega-Redondo<sup>[5]</sup>根据 PathCount 算法挖掘结构洞，他和度中心很相似，节点的得分被定义为当前节点所处于的所有节点对最短路径的均值。

### 1.3 主要研究内容

本文实现了 HIS 算法, ICC 算法, BICC 算法, 而后结合 HIS 算法以及 BICC 算法的优点, 提出了一个新的算法 HIS\_BICC。使用 Twitter 上的数据集来进行性能分析, 分别分析了其不同算法上的 precision, recall, F1\_score。提出的算法在各个方面优于本文介绍的其他算法, 最后的结论的确是这样的, 创新算法在各方面的性能都是不错的。本文最后将结构洞理论与互联网产品运营相结合, 介绍了一款“基于结构洞理论的互联网产品的数据化运营”系统, 如果能够把结构洞理论应用到互联网产品的运营中, 可以将产品的推广直击目标人群, 购买率等将会增加, 这将使公司的获利增加, 同时也能够缩短推广路径, 节约推广成本。我们认为这对于很多公司来说将会是一个不一样的春天, 会为其带来非常多的盈利。

在当今社会中, 社交是互联网、移动互联网上的重要概念与模式。社交网络是一种非常棒的商业模式, 从时间上来讲, 它是推动互联网向现实世界无限靠近的重要角色。社交网络已经深入到人们的生活, 那么如何利用社交网络显得尤为重要。想要利用社交网络就需要找出社交网络中价值比较大的节点, 这样的节点能够在本网络中得到最大化的利益, 这个节点就是结构洞。

结构洞关键技术的应用也是相当广泛的, 在中国多见于网络安全方面的研究, 比如已经有关于情报分析方面的研究。从普遍上来讲, 如果对其中的结构洞节点加以管理控制, 可以有效的达到想要的目标。下面将介绍本文中实现的几个算法, 以及提出的 HIS\_BICC 算法。需要注意的是, 在定义中, 我们只考虑了网络信息而没有考虑内容信息。我们的目标就是对结构洞问题进行理论分析。在挖掘算法中将网络信息和内容信息相组合使用将被留作我们的未来工作之一。

### 1.4 本文组织结构

本文将在下一章介绍研究结构洞的理论基础, 然后第三章介绍了三个结构洞算法, 第四章结合 HIS 算法和 BICC 算法提出创新 HIS\_BICC 算法, 并进行算法对比分析。第五章介绍了结构洞在互联网产品上的应用。

## 第二章 相关研究理论基础

社交网络是一个很广的概念，基于社交网路的研究有很多方面，在这里针我们将研究社交网络中的结构洞关键技术，对于社交网络中的结构洞问题我们需了解一些相关研究的理论基础，为下面章节的算法介绍打好基础。

### 2.1 社交网络

社交网络是个体与个体之间进行交互所形成的整体结构，因此，它本身是一个整体的结构。同时，这个结构是动态变化的，但是，我们在本论文中不会提及社交网络的动态变化，我们仅仅针对某一时刻的社交网络，进而去研究它的图结构。在社交网络中，个体以及个体与个体之间的关系都是其核心，因此，我们在本论文中只是研究其关系未改变的状态。

### 2.2 结构洞

在 1992 年的时候，社会学家罗纳德·伯特提出了结构洞理论的概念,这个概念在社会学领域有着广泛的研究。在社会学中，多年前就已经存在了关于结构洞的一些比较成熟的想法，例如，在社会网络中占领某个位置的人如何从所占据的位置中受益<sup>[6]</sup>。我们从直觉上认为占据不同群体的个人之间的中介或桥梁位置的人，往往可以获得更丰富的信息供应，并且对其网络关系拥有更多的控制能力。这个概念形成了结构洞理论的基础<sup>[7]</sup>。Burt 研究了许多组织的社会结构并介绍了结构洞的概念，认为结构洞就是桥接不同群体的那个位置，并且可以为占有者带来利益好处。它显示了信息在一个单一的社团中往往是同质类似的。非冗余信息的获取通常要通过不同社团之间的联系。

过去十年各种各样的大型网络经历了一个指数级的增长，无论是社交网络还是引用网络，亦或是生物网络，无线网络等。因此，人们开始认真的考虑去开发一些高效的算法以探索这些网络的一些独特性质。结构洞具有非常广泛的应用范围，例如，在社区检测中，识别连接不同社团的中心个体可以帮助隔离和识别社区<sup>[8]</sup>。在流行疾病和谣言蔓延中，隔离结构洞位置的个体可以有效阻止疾病感染的速度和谣言传播到其他社团的速度<sup>[9][10]</sup>。在病毒性营销中，占据结构洞位置的节点的力量是非常强大的，可以通过这些节点迅速占领产品市场，加快新的产品到达不同的社团的速度<sup>[11][12]</sup>。



## 2.3 PageRank 算法

算法源于网页的排序问题，用来评价网页的重要性，基本上可以这样理解：如果网页被多个网页链接到的话，那么这个网页的重要性就比较高，其 pagerank 值也会相对较高，并且由这个网页所链接到的网页的值也高。所以我们一般就是看网页所链接的网页的质量，以及网页被链接的网页的质量。

下面我们将简单的介绍网页排名算法的过程，如图 2-1 所示，可以观察到 B 网页和 C 网页指向了网页 A，我们把  $PR(A)$  定义为跳转到网页 A 的概率：

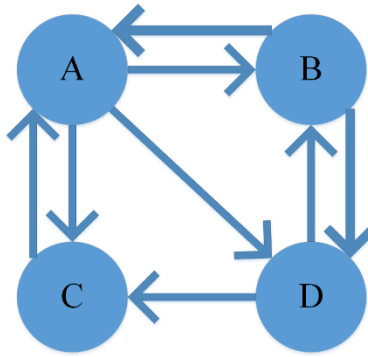
$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1}$$


图 1-1 强连通网页图

通常，用一个矩阵来表示网页之间的跳转关系，如果有  $n$  个网页， $M$  是一个  $n * n$  的转移矩阵，如果网页  $j$  指向  $t$  个网页，那么对每个指向的网页  $i$ ，有  $M[i][j] = 1/t$ 。所以转移矩阵对应如下：

$$M = \begin{pmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix}$$

初始时，我们认为分布在人和一个网页的概率等同，所以初始的概率分布是一个  $n$  维向量  $V_0$ ，所以进行一次网页跳转的概率分布为  $V_1 = MV_0$ 。

$$V_1 = MV_0 = \begin{pmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix} = \begin{pmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{pmatrix}$$

一直迭代下去直至收敛，最终  $V = \begin{pmatrix} 3/9 \\ 2/9 \\ 2/9 \\ 2/9 \end{pmatrix}$ 。但是这样的话，就要求图是强连通图。

而对于图 2-2 来说，由于网页 C 没有出度，当浏览这样的网页的时候会使前面的累

计概率全部清零，最终得到一个零向量。为了满足马尔可夫链的收敛性，我们假设它对所有的网页都有指向。此时我们认为跳转到网页 $A$ 的概率可以定义如下：

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{4}$$

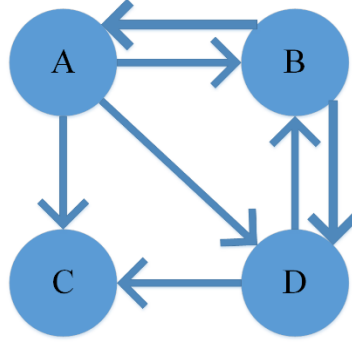


图 2-2 非强连通网页图

然而对于自己只指向自己的网页，或者是几个网页形成环路，如图 2-3 所示。在进行计算的过程中，这些网页的分值是非递减的，这使得这些网页会有超高的概率分很高，但是他网页的概率分布却为零，这种情况是我们需要避免的，因此跳转到网页 $A$ 的概率可以这样定义：

$$PR(A) = \alpha \left( \frac{PR(B)}{2} \right) + \frac{(1 - \alpha)}{4}$$

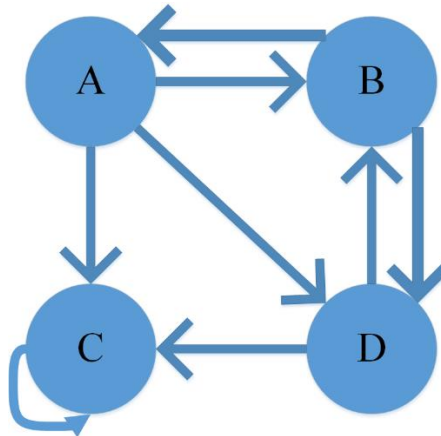


图 3-3 有环网页图

这就是计算网页排名算法的过程。在本论文中，我们用 pagerank 值来表示节点的初始重要性得分。

## 2.4 DFS 算法

深度优先搜索是一个对图进行遍历的算法，它的核心思想是从一个顶点出发，沿着它的一个邻接点一直向下走，直到达到目标点，如果此条路失败，那么就返回上一个节点，重新计算。在本论文中使用了 DFS 算法来计算节点之间的最短路径。

我们将针对图 2-4 中来演示 DFS 过程，从节点 1 开始，首先节点 2 入队列，其次的顺序依次是 3465，因为在搜索过程中只要是当前节点还没有访问过的邻接点，都会被直接加入，知道没有办法再继续向下进行为止，当 6 入队列之后，已经没有节点被访问过了，因此会回退当上一个节点 4，然后发现 4 的一个邻接点 5 还没被访问，所以直接访问了 5，直到所有的节点都被访问了。

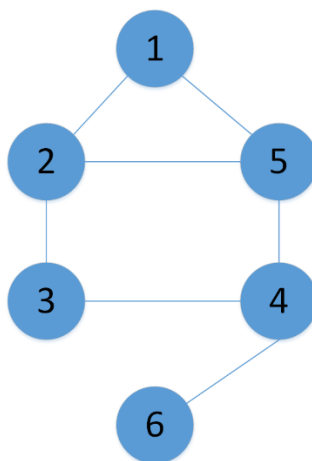


图 4-4 DFS 节点图

## 2.5 本章小结

本章主要介绍了算法所基于的理论基础，包括社交网络以及结构洞的概念，以及算法所基于的 PageRank 算法和 DFS 算法，这为我们后续的研究做了铺垫。

## 第三章 结构洞算法

结构洞算法在社交网络问题的研究中意义重大,对于不同网络基础的算法也衍生出很多的改进,在这里本章节将根据上一章的问题定义,进行算法的研究,分别介绍 HIS 算法、ICC 算法、BICC 算法以及性能分析,同时还将在下一章节对算法进行结果分析及对比。

### 3.1 HIS 算法

#### 3.1.1 基于社团的问题定义

社交网络  $G = (V, E)$ , 其中  $V = \{v_1, v_2, \dots, v_n\}$  是一个包含  $n$  个用户的集合,  $E \subseteq V \times V$  是用户间的无向社交关系集合。进一步假设社交网络中的点可以分成  $L$  个社团  $C = \{C_1, \dots, C_L\}$ , 其中  $V = C_1 \cup C_2 \cup \dots \cup C_L$ 。寻找 top-k 个结构洞问题被定义为寻找一个网络中包含  $k$  个节点的子集, 标记为  $V_{SH}$ 。

基于以上定义,我们认为其背后的常规想法就是衡量节点是如何桥接不同的社团的。我们考虑与当前节点连接的节点对挖掘结构洞的重要性。也就是说,如果一个节点连接了多个重要节点(比如权威性很强的用户),那么这个节点是结构洞的可能性更大。

图 3-1 显示了一个两个社团间的结构洞示例,显然  $v_6$  和  $v_{12}$  可以被看作是两个社团间的结构洞。然而此处我们也存在一些引申问题,例如:(1) 节点  $v_6$  和  $v_{12}$  哪个点占据结构洞的程度更高一些(2) 怎样选择 top-k 个节点集合,使得结果更加准确(3) 后面会讲述我们检测到的结构洞是如何帮助其他的网络应用来完成他们的任务的。

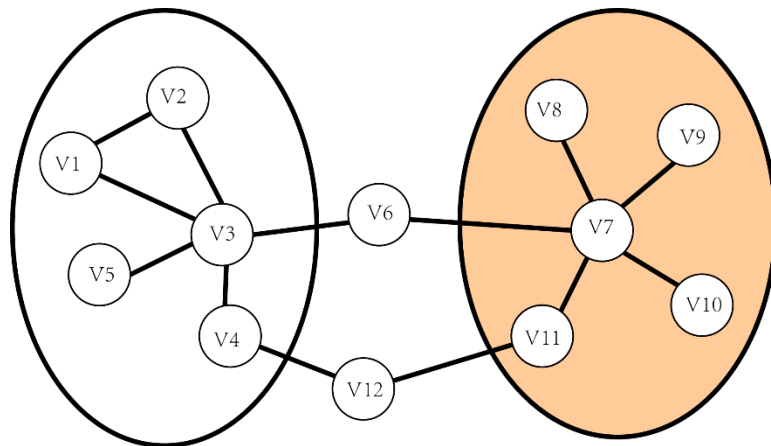


图 3-1 跨两个社团的结构洞

从直觉上来讲，本算法基于两步信息流理论<sup>[13]</sup>，理论认为创新思想首先流向意见领袖，再从意见领袖流向更广泛的人群。从这个意义上讲，用户连接了不同社团中的越多的“意见领袖”，那么用户占据结构洞位置的可能性更高。例如，在图 3-1 中，节点  $v_3$  和  $v_7$  分别充当两个社团的意见领袖，那么他们就比其他节点（如  $v_5$  和  $v_{11}$ ）有更大的能力去传播信息。 $v_6$  和  $v_{12}$  可以被认为是连接两个社团的桥梁，通过比较节点  $v_6$  和节点  $v_{12}$ ，我们看到  $v_6$  连接了两个社团的意见领袖，而  $v_{12}$  只连接了两个普通用户。根据信息流理论， $v_6$  会比  $v_{12}$  具有较高的信息优势（即，较高度度的占据结构洞位置）。另一方面，通过连接结构洞，那么节点可以很自然的增强信息传播的力量。

### 3.1.2 HIS 算法思想

基于前一小节的描述，我们将介绍第一个算法，简称 HIS。HIS 算法是基于社团的，本算法只适用于社团已经给定的情况下。我们首先给出以下定义<sup>[1]</sup>：

给定网络图  $G$ ， $C = \{C_1, \dots, C_L\}$  表示网络图  $G$  中的  $L$  个社团，对于社团的任意子集  $S$ ， $S \subseteq C$ ， $|S| \geq 2$ ，我们定义  $H(v, S) \in [0, 1]$  作为节点  $v$  在  $S$  中的结构洞得分。 $I(v, C_i) \in [0, 1]$  作为节点  $v$  在社团  $C_i$  中的重要性得分。因此，每个节点在每个社团中有一个重要性得分，并且对于每个可能的  $S \subseteq C$  ( $|S| \geq 2$ ) 有一个结构洞得分。两种类型的得分在彼此之间相互递归定义，对于结构洞得分和重要性得分的定义见公式 3-1：

$$I(v, C_i) = \max_{e_{uv} \in E, S \subseteq C \wedge C_i \in S} \{I(v, C_i), \alpha_i I(u, C_i) + \beta_S H(u, S)\} \quad (3-1)$$

$$H(v, S) = \min_{C_i \in S} \{I(v, C_i)\} \quad (3-2)$$

式中  $\alpha_i, \beta_S$ ——是两个可调节参数

用户  $v$  的重要性得分就是用户  $v$  的朋友的重要性得分与结构洞得分的线性组合的最大值。用户  $v$  的结构洞得分就是用户  $v$  在  $S$  中不同社团中重要性得分的最小值。同样的我们可以这样理解，重要性得分可以认为是用户  $v$  从他的朋友接收到的最大信息流，并且  $S$  中的结构洞应该在  $S$  中的所有社团中都活跃。然后根据结构洞的两种类型得分的相互定义递归调用。算法的初始化可以使用 PageRank<sup>[14]</sup> 或者 HITS<sup>[15]</sup> 来计算每个点的权威值  $r(v)$ 。那么重要性得分就可以按照如下方法来定义：

$$\begin{cases} I(v, C_i) = r(v), & v \in C_i \\ I(v, C_i) = 0, & v \notin C_i \end{cases} \quad (3-3)$$

### 3.1.3 HIS 算法分析

HIS 的伪代码如表 3-1 所示：

表 3-1 HIS 算法

| Algorithm HIS  |  |
|----------------|--|
| <b>Input:</b>  | $G = (V, E), \alpha_i, \beta_S, \text{convergence threshold } \epsilon$                      |
| <b>Output:</b> | Importance $I$ and structural hole score $H$   |
| 1:             | Initial $I(v, C_i)$ according to Eq.(3-3)  |
| 2:             | <b>repeat</b> until convergence  |
| 3:             | <b>For</b> each $v \in V$ <b>do</b>  |
| 4:             | <b>For</b> each $C_i \in C$ <b>do</b>  |
| 5:             | $P(v, C_i) = \max_{S \subseteq C \wedge C_i \in S} \{\alpha_i I(v, C_i) + \beta_S H(v, S)\}$ |
| 6:             | <b>For</b> each $v \in V$ <b>do</b>  |
| 7:             | <b>For</b> each $C_i \in C$ <b>do</b>  |
| 8:             | $I'(v, C_i) = \max\{I(v, C_i), \max_{e_{uv} \in E} P(u, C_i)\}$                              |
| 9:             | <b>For</b> each $S \subseteq C$ <b>do</b>  |
| 10:            | $H'(v, S) = \min_{C_i \in S} I'(v, C_i)$   |
| 11:            | Check the $\epsilon$ -convergence condition by   |
| 12:            | $\max_{u \in V, C_i \in C}  I'(v, C_i) - I(v, C_i)  \leq \epsilon$                           |
| 13:            | Update $I = I'$ and $H = H'$   |

在本文中结构洞得分和重要性得分迭代调用，直到达到两个分数的期望的平衡值。实际上， $I(v, C_i)$ 和 $H(v, S)$ 可以无穷大，所以我们给出了存在收敛解的定理。

#### (1) 定理 3.1

对于给定的 $\alpha_i, \beta_S$ ，对任意给定图 $G = (V, E)$ 的重要性得分 $I(v, C_i)$ 和结构洞得分 $H(v, S)$ 总是存在当且仅当

$$\max_{C_i \in C, C_i \in S} \{\alpha_i + \beta_S\} \leq 1$$

证明：

充分性：假设对给定的 $C_i \in C, C_i \in S$ 存在 $\alpha_i + \beta_S > 1$ 。我们考虑两个连接点 $v_1, v_2$ ，其中 $r(v_1) = r(v_2) = 1, v_1 \in \cap_{C_j \in S} C_j, v_2 \in C_i$ 。因此，有 $I(v_1, C_i) = 1$ 。根据 $H(v_1, S) = \min_{C_j \in S} I(v_1, C_j) = 1$ 。因此 $I(v_2, C_i) \geq \alpha_i I(v_1, C_i) + \beta_S H(v_1, S) = \alpha_i + \beta_S > 1$ 就有可能，这与 $I(v, C_i) \in [0, 1]$ 相背，因此假设不成立。

必要性：对于社团 $C_i$ 且 $C_i \in S$ ， $\alpha_i + \beta_S \leq 1$ 。我们可以归纳证明，经过无数次迭代之后，满足 $I(v, C_i) \leq 1$ 。第一次迭代中，有 $I^{(0)}(v, C_i) \leq r(v) \leq 1$ 。经过 $k$ 次迭代之后，有 $I^{(k)}(v, C_i) \leq r(v) \leq 1$ 。因此，在第 $k+1$ 次迭代时，对 $C_i \in S$ ，有

$$I^{(k+1)}(v, C_i) \leq \alpha_i I^{(k)}(u, C_i) + \beta_S H^{(k)}(u, S) \leq (\alpha_i + \beta_S) I^{(k)}(u, C_i) \leq I^{(k)}(u, C_i) \leq 1$$

因此得证。□

根据定义来实现算法的时间复杂度为 $O(K2^L|E|)$ 。其中 $K$ 为迭代次数。下面将讨论它的 $\epsilon$ 收敛，然后再讨论他的效率。

## (2) 定理 3.2

算法满足 $\epsilon$ 收敛，定义 $\gamma = \max_{C_i \in C, C_i \in S} \{\alpha_i + \beta_S\}$ ，则有

$$\max_{v \in V, C_i \in C} |I^{(k+1)}(v, C_i) - I^{(k)}(v, C_i)| \leq \gamma^k$$

证明：参数 $\alpha_i, \beta_S$ 满足 $\gamma = \max_{C_i \in C, C_i \in S} \{\alpha_i + \beta_S\} \leq 1$ 。在迭代过程中，对任意的 $v \in V, C_i \in C$ 且 $S \subseteq C$ ， $I^{(k)}(v, C_i)$ 和 $H^{(k)}(v, S)$ 是关于参数 $k$ 的非递减变量。现在使用归纳证明 $\max_{v \in V, C_i \in C} |I^{(k+1)}(v, C_i) - I^{(k)}(v, C_i)| \leq \gamma^k$ 以及 $\max_{v \in V, S \subseteq C} |H^{(k+1)}(v, S) - H^{(k)}(v, S)| \leq \gamma^k$ 。当 $k = 0$ 时，对任意 $v \in V, C_i \in C$ ，有 $I^{(1)}(v, C_i) \leq 1$ ，因此 $|I^{(1)}(v, C_i) - I^{(0)}(v, C_i)| \leq 1$ 。并且对任意 $v \in V, S \subseteq C$ ，有 $H^{(1)}(v, S) \leq 1$ ，因此 $|H^{(1)}(v, S) - H^{(0)}(v, S)| \leq 1$ 。

假设经过 $k$ 次迭代，对于任意 $v \in V, C_i \in C$ ，有 $|I^{(k+1)}(v, C_i) - I^{(k)}(v, C_i)| \leq \gamma^k$ ，对任意 $v \in V, S \subseteq C$ ，有 $|H^{(k+1)}(v, S) - H^{(k)}(v, S)| \leq \gamma^k$ 。因此，在第 $k+1$ 次迭代中，对于 $v \in V, C_i \in C$ ，如果 $I^{(k+2)}(v, C_i) = I^{(k+1)}(v, C_i)$ ，那么 $|I^{(k+2)}(v, C_i) - I^{(k+1)}(v, C_i)| = 0$ 。否则存在 $u, e_{uv} \in E$ 且 $S \subseteq C, C_i \in S$ 且

$$\begin{aligned} I^{(k+2)}(v, C_i) &= \alpha_i I^{(k+1)}(u, C_i) + \beta_S H^{(k+1)}(u, S) \\ &\leq \alpha_i (I^{(k)}(u, C_i) + \gamma^k) + \beta_S (H^{(k)}(u, S) + \gamma^k) \end{aligned}$$

$$\begin{aligned} &\leq \alpha_i I^{(k)}(u, C_i) + \beta_S H^{(k)}(u, S) + \gamma^{k+1} \\ &\leq I^{(k+1)}(v, C_i) + \gamma^{k+1} \end{aligned}$$

因此, 有  $|I^{(k+2)}(v, C_i) - I^{(k+1)}(v, C_i)| \leq \gamma^{k+1}$ 。对任意  $v \in V, S \subseteq C$ , 存在  $C_i \in S$ , 且

$$\begin{aligned} H^{(k+1)}(v, S) &= I^{(k+1)}(v, C_i) \\ &\geq I^{(k+2)}(v, C_i) - \gamma^{k+1} \\ &\geq H^{(k+2)}(v, S) - \gamma^{k+1} \end{aligned}$$

因此, 有  $|H^{(k+2)}(v, S) - H^{(k+1)}(v, S)| \leq \gamma^{k+1}$

因此, 当  $\gamma = \max_{C_i \in C, C_i \in S} \{\alpha_i + \beta_S\} < 1 - \delta$ , 其中  $\delta$  一个小常量, 算法在有限次迭代后, 被认为是收敛。

现在我们来讨论他的效率, 算法的时间复杂度为  $O(2^L |E| / \log \gamma)$ , 这对于大型社交网络是非常耗时的。我们考虑如何提升算法的性能。注意, 当在第  $k$  次迭代中节点  $v$  的邻居节点的值发生改变, 在算法的  $(k+1)$  次迭代中我们只需计算  $I(v, *)$  的值。我们可以记录在第  $k$  次迭代中每个节点的改变状况, 并在下一次迭代中向他的所有邻居节点广播它的改变。这样我们可以在线性时间内迭代更新。每次迭代, 选择更新值  $I(v, *)$  最大的节点  $v$ , 并向它的邻居广播它的值。如果使用优先级队列, 选择更新值  $I(v, *)$  最大的节点  $v$  可以在  $O(\log |E|) = O(\log |V|)$  内完成。算法运行完成, 我们将选出  $\max_{|S| \geq 2} \{\beta_S H(v, S)\}$  前  $k$  大的  $k$  个节点作为 top- $k$  个结构洞。

## 3.2 ICC 算法

### 3.2.1 基于图结构的问题定义

结构洞桥接不同的社团并且这些社团之间的最短路径会通过结构洞。因此, 这些结构洞节点的移除会增加其它顶点之间最短路径的长度。在图 3-2 中, 节点  $v_1$  在不同社团之间的节点之间的最短路径中扮演了一个重要角色, 并且节点  $v_1$  的移除会明显增加其他节点对之间最短路径的长度, 然而移除其他顶点对最短路径的影响则不是太明显。在这里我们定义前  $k$  个结构洞问题为寻找一个包含  $k$  个节点的集合, 这  $k$  个点的移除将会导致网络的平均距离增加最大化。



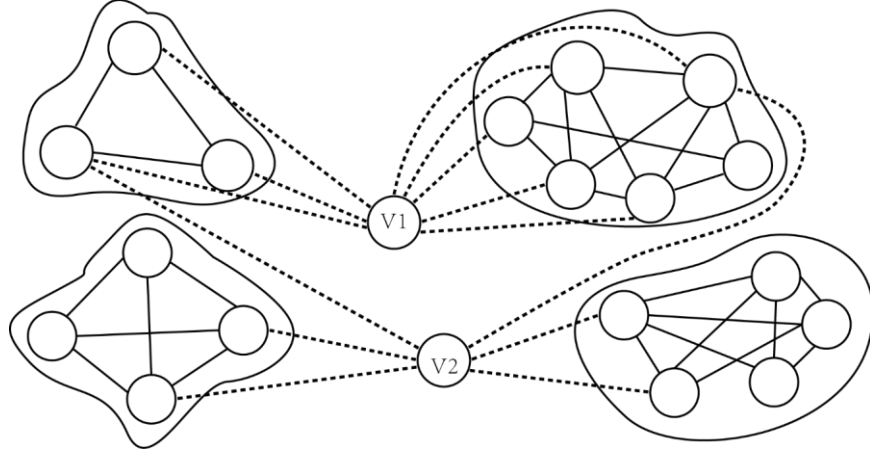


图 3-2 阐明跨多个社团的结构洞

一个社交网络可以看作是一张无向连通图  $G = (V, E)$ ，其中  $V$  为代表独立个体的点的集合， $E$  代表个体之间的关系的边的集合，并且  $|V| = n$ ， $|E| = m$ 。节点  $v$  的度为他的邻居节点的个数，定义为  $deg(v)$ ，图  $G$  中点的最大度定义为  $\Delta(G)$ 。

给定两个节点  $u, v \in V$ 。他们之间的点连通度  $K^G(u, v)$  就是点不连通路的最小数量。如果在图  $G$  中移除不超过  $k$  个点，点  $u$  和点  $v$  仍旧连通，那么他们就被认为是  $k$  点连通的。如果任意一对点都是  $k$  点连通的，那么图  $G$  就是  $k$  点连通的。如果一个点的移除将导致图不连通，那么这个点被认为是关节点，后续研究可以针对关节点进行。

因为图  $G$  是一个无权图，所以我们认为每个  $e \in E$  可以作为一条真实的边，并赋予边权重为 1。图  $G$  中点  $u$  和点  $v$  之间的距离  $d_{uv}^G$  就是它们之间的最短路径。假设对  $v \in V$  有  $d_{vv}^G = 0$ 。给定关于  $V$  的子集  $V_s$ ， $G[V \setminus V_s]$  为图  $G$  中的节点  $V \setminus V_s$  所引导的子图。我们将  $G[V \setminus V_s]$  缩写成  $G \setminus V_s$ 。节点  $v$  在图  $G$  中的逆亲密中心值(inverse closeness centrality)就是节点  $v$  到其它节点的平均距离<sup>[16]</sup>，定义见公式 3-4：

$$c(v) = \frac{\sum_{u \in V} d_{uv}^G}{n-1} \quad (3-4)$$

图  $G$  的平均距离被定义如公式 3-5：

$$c(G) = \frac{\sum_{v \in V} c(v)}{|V|} = \frac{\sum_{v \in V} \sum_{u \in V} d_{uv}^G}{(n-1)|V|} = \frac{\sum_{v \in V} \sum_{u \in V} d_{uv}^G}{(n-1)n} \quad (3-5)$$

那么图  $G$  的所有对最短路径和定义如公式 3-6：

$$C(G) = \sum_{u \in V} \sum_{v \in V} d_{uv}^G = n(n-1)c(G) \quad (3-6)$$

注意：如果图  $G$  是不连通的，图  $G$  的平均距离仍旧有效，不在同一连通分量中的两点的平均距离被定义为充分大的值  $\varphi$  去避免无穷大距离。这个值应该大于图  $G$  中任一连通分量中最短路径长度和。例如  $\varphi = n^3$ ，一个包含  $n$  个点的图的所有对最短路径长度之

和不超过 $\varphi/3$  [17], 问题目标就是:

$$\max_{V_s \subset V, |V_s|=k} \{C(G \setminus V_s)\} \quad (3-7)$$

基于图结构捕获的结构洞的特征如下:

1. 给定个体 $u$ 桥接着多个社团, 另一个个体 $v$ 只和自己社团内部的个体有连接。我们认为个体 $u$ 相对个体 $v$ 占据结构洞位置的可能性更高。因为个体 $v$ 在自己社团内部的个体间连接较强, 个体 $v$ 的缺失只会轻微导致网络中其他个体间的距离增加。相反, 个体 $u$ 连接不同社团中的个体, 因此个体 $u$ 的缺失将显著增加网络中其他个体间的距离, 因为他们之间的连接很松弛。

2. 给定个体 $u$ 桥接较大的社团, 而个体 $v$ 桥接较小的社团, 我们认为个体 $u$ 是比个体 $v$ 更好的结构洞, 因为个体 $u$ 的缺失将导致更多的节点不连通。

3. 给定个体 $u$ 桥接更多的社团, 而个体 $v$ 桥接相对较少的社团, 我们认为个体 $u$ 比个体 $v$ 占据结构洞位置的可能性更高, 因为个体 $u$ 的缺失将会导致更多社团之间的距离增加。

### 3.2.2 ICC 算法思想

我们提到的算法是基于点的有界逆亲密中心值<sup>[3]</sup>(Inverse Closeness Centrality of vertices), 简称为 ICC。

由于社交网络中的一个结构洞通常跨越了多个社团, 结构洞到社交网络中其他点的距离之和通常不大于普通点到社交网络中其他点的距离之和。在移除点 $v \in V$ 后, 网络的平均距离定义见式 3-8:

$$c(G \setminus \{v\}) = \frac{n(n-1)c(G) - 2 \sum_{u \in V} d_{uv}^G}{(n-1)(n-2)} + \frac{\sum_{u, w \in V} (d_{uw}^{G \setminus \{v\}} - d_{uw}^G)}{(n-1)(n-2)} \quad (3-8)$$

在这里如果我们只考虑式子中的第一项作为主要部分, 这就意味着如果节点 $v$ 到其他点的距离越短, 那么移除节点 $v$ 后更有可能最大化图的平均距离。我们使用这个指标来度量图 $G$ 中的前 $k$ 个结构洞。具体做法就是让算法迭代进行。初始时结构洞集合 $V_s$ 为空集, 每一次的迭代, 如果点的逆亲密中心值 $c(v)$ 是 $V \setminus V_s$ 中最小的, 那么一个新的假设的结构洞 $v \in V \setminus V_s$ 被发现并添加到集合 $V_s$ 。过程持续进行, 直到 $V_s$ 中点的数量达到 $k$ , 具体算法伪代码描述如下。

### 3.2.3 ICC 算法分析

ICC 的伪代码如表 3-2 所示：

表 3-2 ICC 算法

| Algorithm ICC  |
|--|
| <b>Input:</b> $G = (V, E), k$<br><b>Output:</b> The set $V_S$ of top-k structural hole spanners<br>1: $V_S \leftarrow \emptyset$<br>2: build a priority queue $Q$ of top k hole spanners with the key of each element in $Q$ is its inverse closeness centrality<br>3: <b>for</b> each vertex $v \in V$ <b>do</b><br>4:     calculate the inverse closeness centrality $c(v)$ of $v$<br>5: <b>if</b> $ Q  < k$ <b>then</b><br>6:         add $v$ to $Q$<br>7: <b>else if</b> $c(v)$ is less than the largest key in $Q$ <b>then</b><br>8:         remove the largest key element from $Q$<br>9:         add $v$ to $Q$<br>10: $V_S \leftarrow Q$ |

算法的主要运行时间用来寻找每个源点  $v \in V$  的单源最短路径树。在图  $G$  中使用广度优先搜索 (BFS) 的时间复杂度为  $O(m + n)$ 。因此，ICC 算法的时间复杂度为  $O(nm + n \log k) = O(mn)$ ，第二部分的  $\log k$  因子是每次优先级队列的优先级操作。尽管这个算法非常有效，但是在包含上百万或上十亿个点的大型社交网络中它的时间复杂度也是相当高的，那么它的时间复杂度是否可以显著提升显然就是一个具有挑战性的问题，比如说是变成线性时间，但是检测结果的质量不受负面影响。下面我们将肯定的回答这个问题，并针对这个问题实现一个更有效的算法。

### 3.3 BICC 算法

#### 3.3.1 BICC 算法思想

算法在 ICC 算法的基础上进行性能提升, 如果节点到图中其它节点的最短距离都很小, 该节点的 Closeness Centrality 值高。因为到其它所有节点的平均最短距离最小, 虽然这些节点可信度不一定有多高, 但是它们却非常热衷于在不同的社团之间传递消息。

现实中的社交网络遵循两个非常重要的事实: 一个就是稀疏性: 每个节点的邻居数量是恒定的, 并且不会随着网络大小成比例增长<sup>[18]</sup>; 另一个就是遵循小世界定律: 任何一对顶点之间的预期距离是一个很小的常数, 它不与网络大小成比例<sup>[19]</sup>。因此, 相对于对网络图 $G$ 中的每个点寻找单源最短路径树, 本算法变成寻找顶点到达给定的邻居层次内的一个局部最短路径树。一个点的邻接点就是它的 level-1 层次邻居, 它的邻接点的邻居就是它的 level-2 层次邻居, 并依此类推。我们称跨越节点 $v$ 的 level- $L$  层次邻居的部分最短路径树叫做(L-bounded shortest tree)  $T_L(v)$ , 点 $v$ 的  $L$  界逆亲密中心值 (L-bounded inverse closeness centrality)被定义为:

$$c_L(v) = \sum_{u \in T_L(v)} d_{uv}^G / (n - 1) \quad (3-9)$$

这里采用和算法 ICC 相似的度量, 本算法和 ICC 算法唯一的不同就是选择前 $K$ 大  $L$  界逆亲密中心值的 $K$ 个点, 而不是 ICC 算法中的选择前 $k$ 小逆亲密中心值的 $k$ 个点, 假设  $K \geq k$ 。其背后的原理可以这样理解: 如果一个点(作为源点)可以在一个小距离比如  $L$  内到达网络中越大部分的点, 那么这个点到其它点的平均距离越短。为了探索顶点之间的多样性以及减少一对邻居同时被选择为 top- $k$  结构洞的可能性。对 top- $k$  个结构洞的候选节点 $K$  可以比 $k$ 大。比如, 可以选取  $K = ck$  ( $c \geq 1$ )。然后计算图 $G$ 中的 $K$ 个点的逆亲密中心值, 并且选择前 $k$ 小的点作为网络的 top- $k$  个结构洞。对于 BICC 算法, 首先识别有界逆亲密中心值 top- $K$  大的前  $K$  个点, 对每个顶点  $v \in V$  使用 BFS 遍历图 $G$ 。假设  $c_L(v)$  是顶点 $v$ 到达 level- $L$  层次邻居内的所有顶点的最短路径的长度和。 $H$  为  $K$  个点的集合, 他们是有界逆亲密中心值 top- $K$  大的点。对每个点  $v \in H$ , 算法使用 BFS 策略计算每个点 $v$ 的逆亲密中心值 $c(v)$ 。最后从选择的  $K$  个顶点中识别  $k$  个顶点, 他们是逆亲密中心值 top- $k$  小的顶点。本算法是基于有界逆亲密中心(Bounded Inverse Closeness Centrality), 简写为 BICC。

### 3.3.2 BICC 算法分析

BICC 的伪代码如表 3-3 所示：

表 3-3 BICC 算法

| Algorithm BICC   |
|--|
| <b>Input:</b> $G = (V, E), k, K, L$<br><b>Output:</b> The set of top-k structural hole spanners $V_S$<br>1: build a priority queue $H$ with the bounded inverse closeness as the key of each element in $H$<br>2: build a priority queue $V_S$ with the inverse closeness as the key of each element in $V_S$<br>3: <b>for</b> each vertex $v \in V$ <b>do</b><br>4:     calculate the bounded inverse closeness centrality $c_L(v)$ of $v$ , using BFS search<br>5: <b>if</b> $ H  < K$ <b>then</b><br>6:         add $v$ to $H$<br>7: <b>else if</b> $c_L(v)$ is larger than the smallest key in $H$ <b>then</b><br>8:         remove the smallest key element from $H$<br>9:         add $v$ to $H$<br>10: <b>for</b> each vertex $v \in H$ <b>do</b><br>11:     calculate the inverse closeness centrality $c(v)$ of $v$<br>12: <b>if</b> $ V_S  < k$ <b>then</b><br>13:         add $v$ to $V_S$<br>14: <b>else if</b> $c(v)$ is less than the largest key in $V_S$ <b>then</b><br>15:         remove the largest key element from $V_S$<br>16:         add $v$ to $V_S$<br>17: <b>return</b> $V_S$ |

给定一个无向连通图  $G = (V, E)$ ，其中最大度是一个常数，对于给定的正整数  $k$  和  $L$ ，对于 BICC 算法来说，时间复杂度为  $O(m + n)$ ，其中  $n = |V|$  并且  $m = |E|$ 。

假设图  $G$  代表了所有节点的邻接列表，首先构造一个部分最短路径树  $T_L(v)$ ，它是一个以  $v$  为根的树。对任意节点  $v \in V$  使用 BFS 策略，时间复杂度： $O(n + m)$ 。然后将识别有界逆亲密中心值前  $k$  大的节点，对于每个节点插入优先级队列  $H$  的时间复杂度为

$O(\log K)$ 。因此对于寻找出优先级队列 $H$ 中的节点将花费 $O(n + m)$ 。从优先级队列 $H$ 中的候选人中识别出优先级队列 $V_S$ 并将选中的候选人节点添加到优先级队列 $V_S$ 中, 将花费 $O(K(n + m) + K \log k)$ 。因此算法的时间复杂度为 $(K(n + m) + K \log k) = O(m + n)$ , 其中 $K = ck$ 是一个常数。因为通常情况下, 在真实的社交网络中, 一个个体的邻居节点通常是一个有限的常数, 它不会随着网路的大小成比例增加。

### 3.4 本章小结

本章主要介绍了社交网络中结构洞关键技术问题的相关问题定义。本章首先对问题定义进行了相关知识的储备, 其次引出社交网络中结构洞关键技术问题中挖掘结构洞的算法, 主要介绍了 HIS 算法、ICC 算法、BICC 算法三个算法。

## 第四章 结合 HIS 以及 BICC 的创新 HIS\_BICC 算法

### 4.1 HIS\_BICC 算法

#### 4.1.1 HIS\_BICC 算法思想

我们假设点集合 $V$ 是一个具有 $n$ 个不同用户的集合，他们来自 $L$ 个不同的群组，表示为： $C = \{C_1, \dots, C_L\}$ ，我们把他们定义为 $L$ 个社团。对每个节点都定义一个重要性得分，一个结构洞得分以及节点的  $L$  界逆亲密中心值，我们用它们来衡量节点作为结构洞的程度。在这里仍然正式定义结构洞问题：寻找 top- $k$  个结构洞。

社交网络 $G = (V, E)$ ，其中 $V = \{v_1, v_2, \dots, v_n\}$ 是一个包含 $n$ 个用户的集合， $E \subseteq V \times V$ 是用户间的关系集合。其中社交网络中的点可以分成 $L$ 个社团 $C = \{C_1, \dots, C_L\}$ ，其中 $V = C_1 \cup C_2 \cup \dots \cup C_L$ 。  $I(v, C_i) \in [0, 1]$  作为节点 $v$ 在社团 $C_i$ 中的重要性得分。对于社团的任意子集 $S$ ， $S \subseteq C$ ， $|S| \geq 2$ ，我们定义 $H(v, S) \in [0, 1]$ 作为节点 $v$ 在 $S$ 中的结构洞得分。根据 HIS 算法找到 $2 * k$ 个候选节点，而后在部分最短路径树中，根据有界逆亲密中心值对节点进行选择，选择在 $L$ 步到达的点更多的节点。

为了考虑的更加全面，改进算法的性能，本算法结合社团以及图结构。对于找出 $k$ 个结构洞，我们遵循这样的原理，首先根据 HIS 算法找出 $K$ 个节点定义为 $H$ ，其中设置 $K = 2 * k$ ，然后对任意 $v \in H$ ，计算节点的有界逆亲密中心值(L-bound inverse closeness centrality)，构造一个优先级队列 $V_S$ ，对于任意 $v \in H$ ，前 $k$ 个直接加入 $V_S$ ，而后对节点的有界逆亲密中心值和结构洞得分的线性组合与 $V_S$ 中最小的组合做对比，如果节点的值较大，那么将会入队。

我们在这里也定义了节点的重要性得分以及结构洞得分，并且跟 HIS 算法中为节点定义的重要性得分以及结构洞得分保持了一致，因此，我们选择的结构洞节点首先是在社团中保持了活跃状态，而后根据 BICC 算法 计算了节点的  $L$  界逆亲密中心值，保证了节点能够在给定步长内达到更多的节点，也就是说节点能够在更短的距离内到达更多的节点。

### 4.1.2 HIS\_BICC 算法分析

HIS\_BICC 的伪代码如表 4-1 所示：

表 4-1 HIS\_BICC 算法

| Algorithm HIS_BICC  |
|---|
| Input: $G = (V, E), \alpha_i, \beta_S, \text{convergence threshold } \epsilon, k, K, L$                                   |
| Output: The set of top-k structural hole spanners $V_S$   |
| 1: build a set $H$ with the vertex as the key of each element in $H$  |
| 2: build a priority queue $V_S$ with the L-bound inverse closeness centrality as the<br>key of each element in $V_S$      |
| 3: Initial $H$ with HIS algorithm's top-K structural holes vertex   |
| 4: for each $v \in H$ do  |
| 5:     calculate the L-bound inverse closeness centrality of $v$ as $c(v)$ and<br>structural hole score of $v$ as $sh(v)$ |
| 6:     if $ V_S  < k$ then  |
| 7:         add $v$ to $V_S$   |
| 8:     else if $\alpha * sh(v) + (1 - \alpha) * c(v)$ is less than the largest key in $V_S$ then                          |
| 9:         remove the largest key element from $V_S$  |
| 10:        add $v$ to $V_S$   |
| 11: return $V_S$  |

算法的时间复杂度为 $O(\frac{2^L|E|}{\log \gamma} + n + m)$ ，它首先执行 HIS 算法选出  $K$  个节点，并放入优先级队列 $H$ 中，时间花费 $O(\frac{2^L \log |E| \log K}{\log \gamma})$ ，从  $K$  个候选节点中选择前  $k$  个有界逆亲密中心值和结构洞得分大的  $k$  个节点放入优先级队列 $V_S$ 中将花费 $O(\frac{2^L|E|}{\log \gamma} + n + m)$ ，基本和 HIS 算法的时间复杂度相差不大。

为了验证 HIS 算法、ICC 算法、BICC 算法以及 HIS\_BICC 算法的运行效率以及查准率，查全率，F1\_score 值，本章利用了一组测试数据的运行结果，并对其进行了比较讨论，这里使用 Twitter 的数据集，通过对比分析各个算法的运行效率来探究各种算法的优缺点；与此同时，本章对所提出的 HIS\_BICC 算法的运行效率和查准率查全率，



F1\_score 值进行了评估。

## 4.2 实验环境搭建与测试

### 4.2.1 测试环境描述

1. 操作系统: windows 10 , Linux, 64bit 操作系统
2. windows 电脑配置: Core i3 处理器, 4GB RAM
3. 开发工具: Sublime Text, PyCharm
4. 安装 npm 环境, Flask 框架以及 D3js 库
5. Linux 电脑配置:

Linux version 3.10.0-514.2.2.el7.x86\_64 ([builder@kbuilder.dev.centos.org](mailto:builder@kbuilder.dev.centos.org))

(gcc version 4.8.5 20150623 (Red Hat 4.8.5-11) (GCC) ) #1 SMP Tue Dec 6 23:06:41 UTC 2016

### 4.2.2 实验测试数据

我们使用 Twitter 数据集对实现的各个算法进行测试。实验数据从 Twitter 中选取了 92180 个节点以及 188971 条边, 数据从<sup>[1]</sup>中获取, 见表 4-2。随着算法选择的 top-k 个结构洞的  $k$  值的变化, precision 值, recall 值, F1\_score 值都会跟随变化。随后我们也会根据伯特定义的结构洞指标来衡量挖掘的结构洞。

表 4-2 数据集

| Dataset | Vertex | Edge  |
|---------|--------|-------|
| Twitter | 92180  | 18897 |

## 4.3 算法性能对比分析

下面所得到的结果都是关于以上章节所写算法的运行结果, 都是基于 Twitter 数据集进行测试和评估的。通过这几个算法对 Twitter 的测试数据进行的详细的测试, 本节将全面的对各算法所对应的运行时间, 查准率, 查全率, F1\_score 以及伯特的结构洞进行了记录。为了方便观测, 我们使用 Twitter 中的较小数据集, 所有运算时间均为秒级。

HIS 算法是基于社团的, 在这里, 我们仅展示了在社团 1, 2, 3 中的结构洞。通过

前人大量的实验,我们沿用前人的方法,在此处设置 $\alpha_i = 0.3$ 以及设置 $\beta_s = 0.5 - 0.5^{|S|}$ 。

BICC 算法基于图结构, 首先选择 $K$ 个能够在 $L$ 内到达更多个节点的点, 然后对这 $K$ 个节点中的每个节点计算其到图中其他所有节点的平均距离。算法中设置 $K = 2 * k, L = 4$ 。

### 4.3.1 算法运行时间比较

ICC 和 BICC 的运行时间如图 4-1 所示:

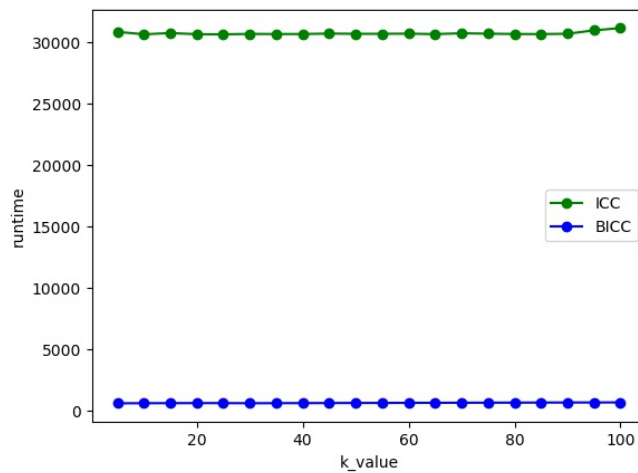


图 4-1 ICC 和 BICC 运行时间比较

因为 ICC 算法的时间复杂度较高, 我们首先来看 ICC 和 BICC 算法的运行时间从图中可以看出 ICC 算法的运行时间远远超过了 BICC 的运行时间, BICC 算法相对于 ICC 算法的优化是非常成功的。

HIS, BICC 和 HIS\_BICC 的运行时间如图 4-2 所示:

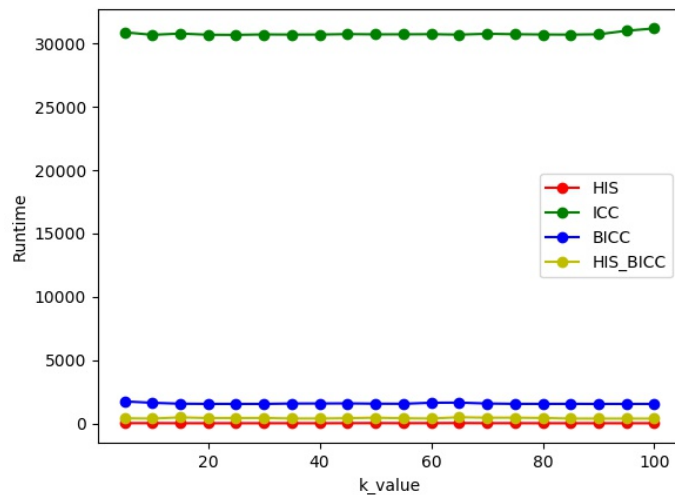


图 4-2 算法运行时间比较

我们提出的 HIS\_BICC 算法，结合了 HIS 算法的优点，算法的运行时间基本和 HIS 算法相当，但是比 BICC 算法的运行时间要低。这可以更快的帮助应用，减少算法的等待时间。

### 4.3.2 算法查准率比较

HIS, ICC, BICC 和 HIS\_BICC 的查准率比较如图 4-3 所示：

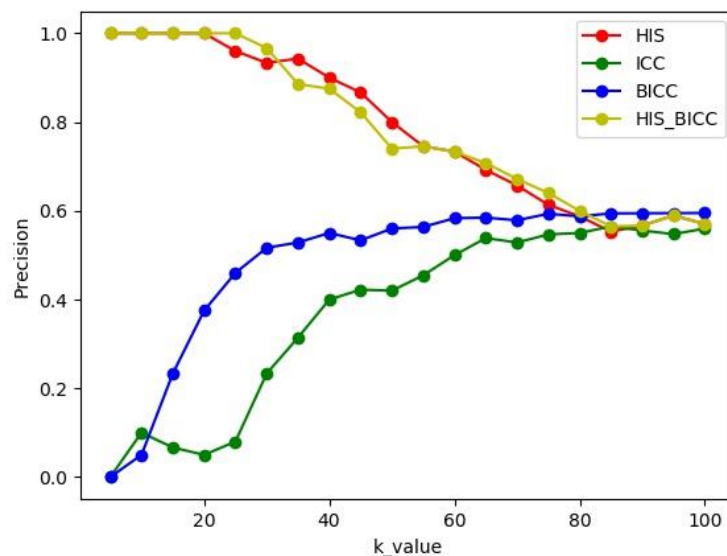


图 4-3 算法查准率比较

随着  $k$  值的增加，ICC 及 BICC 的查准率也增加，HIS 的查准率开始下降，我们提

出的 HIS\_BICC 算法自然也在下降，但是它的查准率在某些值上是高于其他的算法的，因此，我们认为我们提出的 HIS\_BICC 算法在查准率上来讲算是较优的。

### 4.3.3 算法查全率比较

HIS, ICC, BICC 和 HIS\_BICC 的查全率如图 4-4 所示：

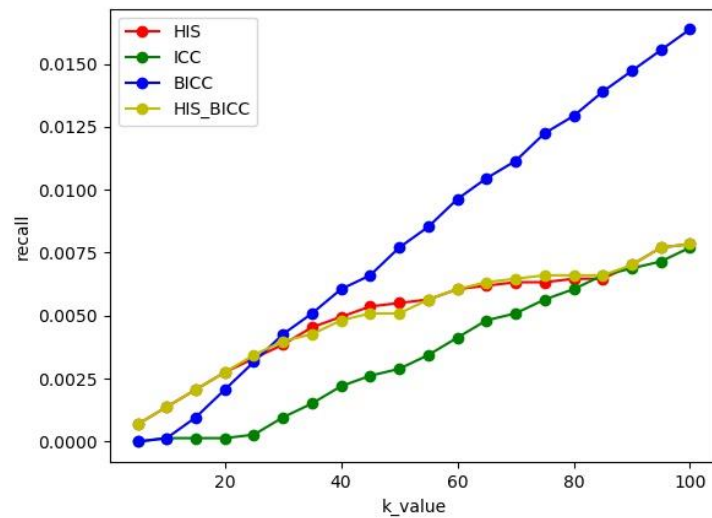


图 4-4 算法查全率比较

所有算法的查全率都会随着  $k$  值的增加而增加，这是因为  $k$  值增加了，相对来说，找到的结构洞更全一些，这是跟查全率的定义有关系的。我们可以明显的看到我们提出的算法的性能在查全率上来讲处于中上水平，虽然没有 BICC 高，但是已经超越了 ICC 以及 HIS 算法。

### 4.3.4 算法 F1\_score 比较

HIS, ICC, BICC 和 HIS\_BICC 的 F1\_score 值如图 4-5 所示：

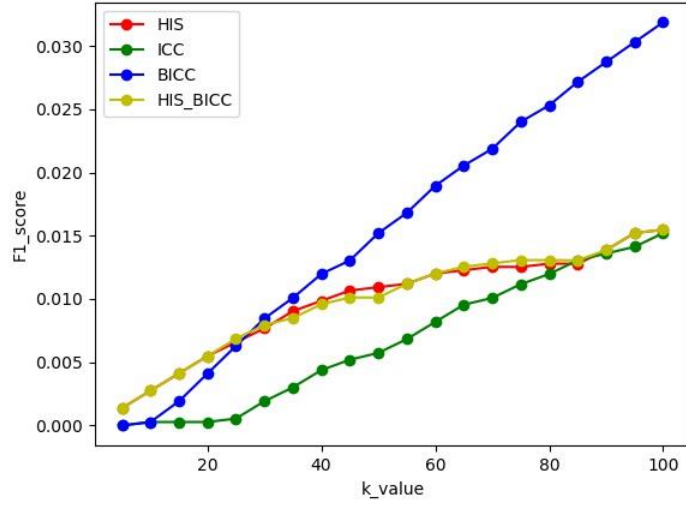


图 4-5 算法 F1\_score 比较

对于算法的查准率和查全率的综合比较，从上图可以看出我们提出的算法的 F1\_score 值仍旧处于中上水平，与查全率基本保持一致。

#### 4.3.5 伯特的结构洞指标

我们首先使用伯特的定义来评估算法的性能，(1)每个个体跨越的社团的大小(2)社团的数量(2)个体的邻居数量。伯特认为一个优质的结构洞应该连接着很多社团，但是为了更有影响力，我们认为节点跨越的社团与节点的邻居的比值应该很大。这种定义实现了在给定社团的情况下，对结构洞的评估标准。给定图  $G = (V, E)$ ，假设  $S$  由算法发现的结构洞的集合。所以找到的结构洞  $S$  集合的质量为：

$$\rho(S) = \frac{\sum_{v \in S} \frac{\# \text{ of communities that } v \text{ is connected to}}{\deg(v)}}{|S|}$$

HIS, ICC, BICC 和 HIS\_BICC 的伯特结构洞性能如图 4-6 所示：

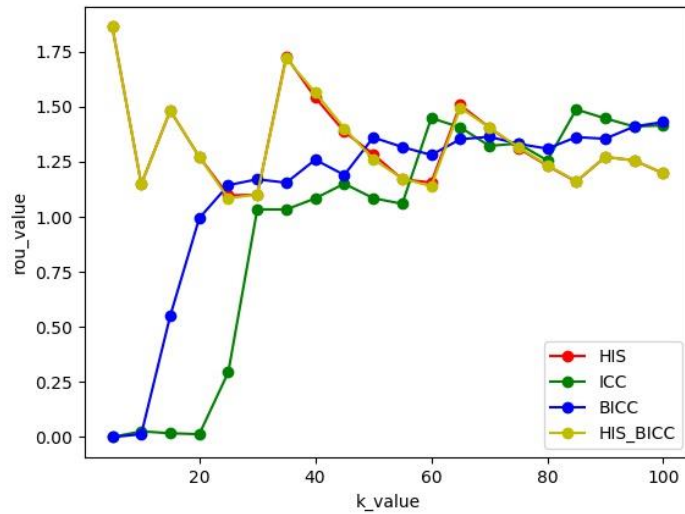


图 4-6 算法结构洞性能比较

根据伯特对结构洞的定义，当  $k$  值越来越大时，ICC 及 BICC 算法表现出更好的性能，在伯特的结构洞度量上来说，我们提出的算法在  $k$  值较小时比较棒。

#### 4.3.6 算法性能综合分析

我们提出的算法结合了基于社团的 HIS 算法以及基于结构的 BICC 算法的优点，考虑了图结构和社团，从而将会从某些方面优于某些算法，事实证明确实是如此，我们提出的算法从某些方面来讲的确是优于其他的算法的。在这里我们将我们提出的 HIS\_BICC 算法和实现的其他算法进行比较，我们从算法的运行时间，查准率，查全率，F1\_score 值以及伯特的结构洞等方面对算法进行评估。由于 BICC 算法和 ICC 算法仅仅基于图结构，所以直接与基于社团的算法进行比较其实是很不公平的。在这里，我们依旧直接单纯的进行了比较。

我们提出的算法是基于 HIS 算法以及 BICC 算法的，虽然算不上很大的创新，但是从各个方面上来讲的确是优于其他的算法的。所以我们初步认为我们提出的 HIS 算法和 BICC 算法结合的 HIS\_BICC 算法是非常好的。但是算法由于算法是在 HIS 算法的基础上的，所以需要社团是提前已知的，这对于大型社交网络来说是非常耗时的，因此使用本算法也是需要慎重考虑的。

## 4.4 本章小结

本章为本论文的核心部分，主要不仅展示了基于社团的 HIS 算法、基于图结构的 ICC 算法、BICC 算法、以及一个创新算法 HIS\_BICC 算法，针对于 Twitter 数据集进行的实验评估测量，分别从查准率，查全率，F1\_score 值和运行时间方面分析了几个算法的性能以及优缺点。并且描述了 HIS\_BICC 算法的伪代码，实现过程以及时间复杂度分析，还介绍了 HIS 算法、ICC 算法、BICC 算法、以及 HIS\_BICC 算法的核心代码。

## 第五章 结构洞在互联网产品上的应用

结构洞可以应用在社团检测，病毒营销，link 预测，传播控制，网络安全等。所以挖掘网络中的结构洞是非常重要并且非常有价值的。其在中国的应用大多数集中在情报分析领域，现在国内在情报分析领域已有大量的研究，对于互联网产品方面则从未出现，在本论文中，我们将假设针对一家农民专业合作社的千村电商平台(以下简称千村)作为案例，验证结构洞理论在互联网产品运作中的作用。此处将构建一个“基于结构洞理论的互联网产品的数据化运营”应用。在互联网产品的运营中，把结构洞理论应用在互联网产品的运营里，可以缩短所推广的信息达到用户的途径，使产品的信息高效的在目标客户中传播，还可以使推广信息到达真正的目标客户。

本应用借助 Python 语言实现了算法，根据流行的 JavaScript 库 D3js 来实现可视化研究，基于 Flask 框架来实现整个应用的管理工作。下面我们将在以下几个方面介绍此应用：

### 1、结构洞在互联网产品上的应用背景

在移动互联网时代，运营是许多网络公司的最重要的事情。运营的目的在于找到目标客户，激活目标客户，保留目标客户，促使目标客户消费，降低公司的业务成本。传统的互联网产品运营措施主要是靠补贴，地推，交换流量，病毒式推广等方式争取流量，在大量用户中“偶遇”有真正需求的用户。这些推广方式仅仅能瞄准大概的目标客户，推广成本高。2015 年大电商的新客获取成本高达 400 元人民币每位，垂直品类电商高达 150 元人民币。在互联网产品的运营中，把结构洞理论应用在互联网产品的运营里，可以缩短所推广的信息达到用户的途径，使产品的信息高效的在目标客户中传播，还可以使推广信息到达真正的目标客户。我们在此处的目标就是减少推广成本，并将推广物品直击目标人群。

千村是一家针对农民的农资电商平台，是一家互联网创业公司。目前，各大电商平台都在向农村扩展业务，主要有阿里农村电商和京东农村电商，但是千村基于本土化的推广方式，在本地推广效率却高于阿里和京东的农村电商。我们分析，千村的推广方式是基于本地熟人的推广方式，存在着个体与个体之间的关系，与结构洞理论相契合。我们认为，如果把结构洞理论深入应用到千村的运营当中，用结构洞理论分析千村的销售数据，以理论去指导千村的运营，推广效率还会提高。



## 2、受众人群

应用针对运营部门的人员，运营人员可以在本应用中分析结构洞与其他人的关系，并针对这些结构洞位置的人群进行推广，这将减少推广路径，节约成本，对特定人群的推广也会增加公司销量，使得获利增加。对运营人员来说，如果从大量的客户中去寻找目标人群，这无疑是非常吃力的，并且效果不见得会有多好。但是当使用本系统后，运营人员则可以有针对性的对结构洞位置的人进行推广，并且也能有效的获取新客，当结构洞位置的人群对我们的发布信息进行转发后，对于这些结构洞位置的人群来说，可能会为他的朋友圈提供很多的资源，那么它们的朋友圈中的人就会对我们的平台进行关注，当然也有可能产生购买行为，这无形中就会增加盈利。

## 3、开发难点

本应用的开发难点在于数据的获取，对于公司来说，这些数据一般情况下是很难获取的。所以我们将对数据进行模拟，由于大规模的数据量运行起来是非常耗时的，为了方便观察，我们的模拟的数据量很小，仅仅观测了 100 个节点，因此从观察效果上来讲，可能并不是那么的直观，但是已足以让运营人员对图结构有一定的了解，并针对性的进行推广。

## 4、研究路径

我们的应用旨在运用结构洞理论来指导运营部门高效工作，从而创造更大的利润。我们希望能够运用有效的知识解决更多的问题。由于数据获取难度偏大，因此本系统的数据是一个虚构的网络，我们认为如果想要一个真实的数据集可以采用下面的方式。通过分析千村的推广路径数据，找到千村客户中的结构洞用户。因为平台积累了大量的微信服务号关注用户，可以通过微信后台可以找到转发量有关的数据。(1)分析结构洞位置的用户与用户所在地区销售量的关系(2)运营部门针对结构洞位置的用户做定向营销(3)比较定向营销之后的获客量，日活量，留存率，消费频次和销售额得到结构洞理论与互联网产品运营的相关结论。

## 5.1 应用分析

应用界面显示了用户与用户之间的关系，近年来，数据可视化越来越流行，很多媒体、新闻、门户网站等都使用了可视化技术，这使得人们更好的理解复杂的数据。所以我们首先对整体的结构进行了显示，使得运营人员清楚的认识个体之间的关系。其次对

于算法显示界面，我们使用了 D3(Data-Driven Documents)来显示挖掘出的结构洞。系统旨在为后台运营人员寻找推广人群，所以本应用显示出各个算法的运行结果，为了更好的理解，我们还将数据进行了可视化，方便了运营人员更好的理解。

图 5-4 的点代表了一个真实的个体，在互联网产品应用中我们可以理解为这个个体主要关注一种种子，比如玉米，小麦，花生等。点与点之间的连线代表了个体之间的关系，可以理解为个体与个体之间存在着好友关系，在我们的应用中可以明显的看到节点是具有群聚效果的，也就是说节点与节点之间是可以以社团来进行划分的，我们在应用中对节点进行了社团划分，并针对 HIS 算法、ICC 算法、BICC 算法以及 HIS\_BICC 算法进行了结构洞的挖掘与显示，用不同颜色的点来表示结构洞节点以及非结构洞节点。

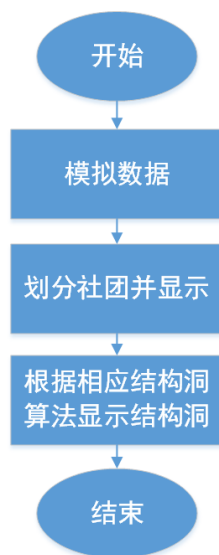


图 5-1 应用基本流程图

图 5-1 是本应用的基本流程图，我们首先运用 Python 的 networkx 库内置的算法来模拟数据集，然后对数据进行社团划分，并在网站首页显示社团结构。当运营人员想要 HIS 算法下运行时可以点击相应位置来进行运行了 HIS 算法之后的结构洞显示图，其他算法也是，想要得到哪个算法下的结构洞显示图，直接点击即可。



图 5-2 系统首页

图 5-2 显示了应用的首页，一张大图代表了我们这个应用是一个关于社交网络的图，其中在右上角设置了登陆按钮，实现了登陆功能。

图 5-3 系统登陆界面

图 5-3 显示了应用的登陆界面，因为是运营部门专门用来看图，所以在这里设定了预置的用户名以及密码。

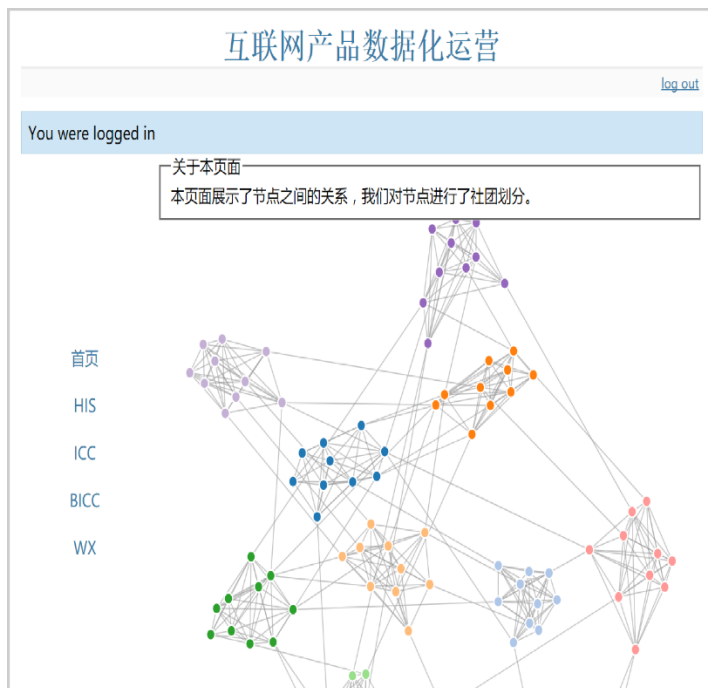


图 5-4 社团划分页面

图 5-4 显示了应用的划分社团之后的页面，其中每个颜色代表了一个社团，这可以让我们清晰的了解节点所属于的社团。

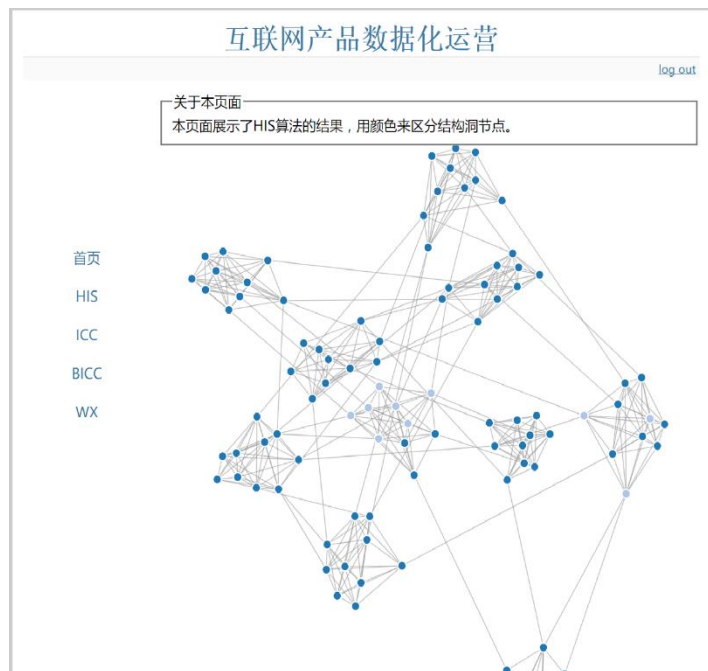


图 5-5 找出结构洞

图 5-5 显示了运行算法之后找出的结构洞，其中共有两种颜色，这里仅仅找出了是个结构洞，显然，浅蓝色代表了这十个结构洞。

## 5.2 预期结果

我们认为结构洞理论是和这种互联网产品相契合的，通过针对本应用给出的结果进行定向推广，我们认为我们所推广的用户所在地区的销售量可以有明显的上升趋势，所推广用户地区的获客量会增加明显，日活量也变得越来越，留存率也会有明显提高，单个用户的消费频次会明显增加，销售额更是会有明显提升。因此结构洞关键技术会对整个公司产生巨大的盈利。

## 结论

本论文对 HIS 算法, ICC 算法, BICC 算法进行了实现, 对其进行性能分析, 研究其 precision, recall, F1-score 值。其中 HIS 算法的运行时间以及精确度等性能都较优秀一些, 因为此算法需要社团是提前给出的, 所以对于没有给出社团的网络, 则需要自己来划分社团, 但是社团的划分是非常耗时的, 因此, 直接比较这几个算法是不公平的。当然如果社团已经给定了, 那么 HIS 算法还是非常优秀的。对于 ICC、BICC 算法, 它们并不适合  $k$  值较小的情况, 运行时间对于大规模的网络来说也相对较长。本论文结合算法的优点, 进而提出自己的算法 HIS\_BICC, HIS\_BICC 算法是在 HIS 算法的基础上来计算节点在网络中的平均距离。所以 HIS\_BICC 算法也是适合于社团给定的情况。

随后, 我们考虑将结构洞应用到互联网产品中, 所以此处构建了一个系统“基于结构洞理论的互联网产品的数据化运营”。据我们了解, 这是首次将结构洞应用到互联网产品的运营中, 对于此系统, 可以有效的针对人群进行推广并且降低获取新客的成本, 比较定向营销之后的获客量, 日活量, 留存率, 消费频次和销售额, 我们认为将结构洞理论与互联网产品运营的相结合, 可以有效的降低获客量, 提升日活率以及留存率, 消费频次, 并且销售额也会提升。因此我们认为结构洞理论与互联网产品运营非常契合, 缩短了中间的推广路径, 直达目标群体。因此, 我们认为如果把结构洞理论应用到互联网产品的运营中, 将会使得公司的盈利增加, 极大的降低推广成本。

## 参考文献

- [1] Lou T, Tang J. Mining structural hole spanners through information diffusion in social networks[C]// The International Conference. 2013:825-836.
- [2] Zhang E, Wang G, Gao K, et al. Generalized Structural Holes Finding Algorithm by Bisection in Social Communities[C]//International Conference on Genetic and Evolutionary Computing. IEEE, 2013:276-279.
- [3] Rezvani M, Liang W, Xu W, et al. Identifying Top- k, Structural Hole Spanners in Large-Scale Social Networks[C]// ACM International on Conference on Information and Knowledge Management. ACM, 2015:263-272.
- [4] Page L. The PageRank citation ranking : Bringing order to the web[J]. Stanford Digital Libraries Working Paper, 1998, 9(1):1-14.
- [5] Bhowmik T, Niu N, Singhanian P, et al. On the Role of Structural Holes in Requirements Identification:An Exploratory Study on Open-Source Software Development[J]. Acm Transactions on Management Information Systems, 2015, 6(3):1-30.
- [6] V, Rijt A V D. Dynamics of Networks if Everyone Strives for Structural Holes[J]. American Journal of Sociology, 2008, 114(Volume 114, Number 2):371-407.
- [7] R.S.Burt.Structural Holes:The Social Structural of Competition[M].Harvard University Press.1992.
- [8] Wang L, Lou T, Tang J, et al. Detecting Community Kernels in Large Social Networks[C]// IEEE, International Conference on Data Mining. IEEE, 2011:784-793.
- [9] Budak C, Agrawal D, Abbadi A E. Limiting the spread of misinformation in social networks[C]// International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April. DBLP, 2011:665-674.
- [10] Guille A, Hacid H, Favre C, et al. Information diffusion in online social networks: a survey[J]. Acm Sigmod Record, 2013, 42(2):17-28.
- [11] Tang J, Wu S, Sun J. Confluence: conformity influence in large social networks[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2013:347-355.
- [12] Yang Y, Tang J, Leung C W, et al. RAIN: Social Role-Aware Information Diffusion[C]// 2015.
- [13] Katz E. The Two-Step Flow of Communication: An Up-To-Date Report on an Hypothesis[J]. Public Opinion Quarterly, 1957, 21(1):61-78.
- [14] Page L. The PageRank citation ranking : Bringing order to the web[J]. Stanford Digital Libraries Working Paper, 1998, 9(1):1-14.
- [15] Kleinberg J M. Authoritative sources in a hyperlinked environment[C]// Acm-Siam Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 1998:668-677.
- [16] Beauchamp MA. An improved index of centrality.[J]. Behavioral Science, 1965, 10(2):161.
- [17] Plesník J. On the sum of all distances in a graph or digraph[J]. Journal of Graph Theory, 2006, 8(1):1-21.

- [18] Barabási A L Emergence of Scaling in Random Networks[J]. Science, 1999, 286(5439):509.
- [19] Kleinberg J M. Navigation in a small world[J]. Nature, 2000, 406(6798):845.



## 致谢

非常感谢谭龙老师在整个毕业设计过程中对我的帮助，过程中一直督促我踏下心来，认真学习，最终也不负期望，成功的完成了毕业设计。在这个过程中经历了很多困难，我想我学会了很多东西，不仅仅是论文的完成，还有面对困难迎难而上的勇气。这个过程让我更加相信我自己，让我充分的肯定了自己。非常感谢在这个过程中给予我帮助的那些人，相信这个世界阳光明媚，这些人们赤诚善良，我们最终都会迎来美好的明天。