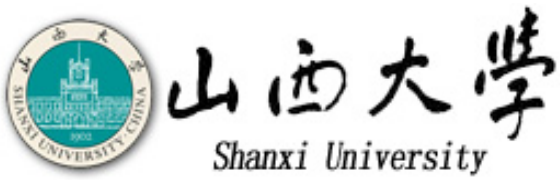


2016 届 硕 士 学 位 论 文

基于密度聚类的网络社区发现算法

作者姓名　　赵　越

指导教师　　白　亮 副教授

学科专业　　计算机技术

研究方向　　网络数据挖掘

培养单位　　计算机与信息技术学院

学习年限　　2014 年 9 月至 2016 年 6 月

二〇一六年六月

山西大学

2016 届硕士学位论文

基于密度聚类的网络社区发现算法

作者姓名　　赵　越

指导教师　　白　亮 副教授

学科专业　　计算机技术

研究方向　　网络数据挖掘

培养单位　　计算机与信息技术学院

学习年限　　2014 年 9 月至 2016 年 6 月

二〇一六年六月

Thesis for Master's degree, Shanxi University, 2016

Community Discovery Algorithms for Networks Based on Density Clustering

Student Name	Zhao Yue
Supervisor	Prof. Bai Liang
Major	Computer Technology
Specialty	Data Mining for Networks
Department	School of Computer and Information Technology
Research Duration	2014.09-2016.06

June, 2016

目录

Contents

中文摘要

社区发现是网络数据挖掘的一个重要研究内容，被用来探索网络中潜在的类结构。它能够帮助人们识别特殊网络群体，选择营销策略，进行产品推荐等，已被广泛应用于生物网络、社交媒体、商业交易和物流运输等。本文以网络数据为研究对象，基于密度聚类技术，对全局和局部社区发现进行了系统地研究，主要研究内容如下：

（1）针对网络数据，本文对密度算法 CFSFDP 进行了扩展，提出了新的节点密度和相异性度量，使得该算法能够有效应用于网络社区发现中。新扩展的算法不仅继承了原算法能够发现任意形状的类结构的优点，而且能有效快速地处理网络数据。最后，通过在大量的真实网络数据和合成网络数据上的实验分析，展示了新算法的有效性。

（2）随着网络数据的规模不断增大，全局社区发现的计算成本是高昂的。此外，在很多情况下，用户往往更加关注于局部社区。因此，基于第一个研究内容，本文提出了一种局部搜索算法，即通过一个给定的节点，去搜索它所在的社区中心点和边界点，从而快速发现该社区。实验结果展示了新算法能够快速而有效地发现网络中的局部社区。

（3）设计并开发了一个社区发现实验系统，该系统包括数据导入，算法选择，评价函数选择，社区发现结果显示等功能，系统中集成了本文提出的算法和传统算法，可以对真实网络数据和合成网络数据进行测试。该系统有着较好的可用性和扩展性。

本文的研究成果进一步丰富了网络社区发现方面的研究，为网络数据挖掘与知识发现相关领域提供了新的技术支撑。

关键词：复杂网络；社区发现；密度聚类；局部社区

ABSTRACT

Community detection is an important research content of data mining for networks, which is used to explore the potential of cluster structure in the networks. It can help people to identify specific network groups, select the marketing strategies, and product recommendations, etc. It has been widely used in biological networks, social media, commercial transactions and transportation of logistics. The object of research is the network data. In the paper, we make a systematic study based on density clustering technology for global and local communities. The main contents of this thesis are summarized as follows:

(1) For network data, the density algorithm is proposed which is an extension of CFSFDP. We propose the new density and dissimilarity measures for nodes so that the algorithm can be applied to the community detection. The new extended algorithm not only inherits the advantages of the original algorithm, but also can find the cluster structure of any shape. Besides, it can quickly and effectively handle network data. Finally, the experimental analysis on a large number of real network data and network data shows the effectiveness of the new algorithm.

(2) As the scalability of the network data increases, the cost of global community detection is very expensive. In addition, in many cases, users tend to focus on the local community. Therefore, based on the content of the first study, we propose a local search algorithm which begins from a given node to search for the center and the border points of its community. Experiments demonstrate that this algorithm can efficiently find the local communities in the network.

(3)We design and develop an experimental system for community detection. This system includes the data importation, the algorithm selection, the evaluation function selection, community detection. The system integrates the traditional algorithms and the new algorithms proposed in this

paper. The tested network data includes the real and synthetic networks. The system has good availability and scalability.

The above mentioned contributions has further enriched the research on the community detection, and provide a new technology support for the studies of the related fields on the network data mining and knowledge discovery.

Key words: Complex networks; Community detection; Density clustering; Local community

第一章 绪论

1.1 社区发现算法的研究意义

在真实世界中有着大量的数据以复杂网络的形式存在，如社交网络，蛋白质网络和物流交通网络等。复杂网络具有小世界性质(Small world)，无标度性质(Scale free)，以及宏观结构特征（Macroscopic）[1, 2]。社区结构是复杂网络的一个重要特性。因此，社区发现获得了众多学者的广泛关注，并成为了网络数据挖掘的一个重要技术手段。在一个网络中，社区往往具有如下特征：社区内部节点之间联系紧密，且节点性质相似，与社区外部其它节点联系较为松散[3]。由于复杂网络的拓扑结构反映了网络中节点之间的关联信息[4, 5, 6]，因此，如何利用该拓扑信息获得网络的类结构是社区发现的重要研究内容。社区发现的目的在于帮助人类理解复杂网络，通过对网络拓扑关系的研究和解释，人们能更好地划分社区结构，并能够从复杂网络中获得隐藏信息。同时，我们还可以预测网络中用户的行为和爱好，根据预测结果进行社区推荐等。由此看来，社区发现具有重要的实用价值。

1.2 社区发现算法的研究现状

社区发现算法发展经过了漫长的历史时期，获得国内外学者的广泛关注。为了更好地解决社区发现的相关问题，多种算法陆续被提出，大致分为以下几类：

图分割算法[7, 8]是社区发现的一个重要技术手段，它将社区发现问题看作为一个图分割[9, 10]问题，实现网络节点的划分。谱聚类(Spectral clustering)算法[11]是其代表性算法之一。该算法基于谱图理论基础，利用拉普拉斯相似性矩阵[12]的特征分解，获得图的新特征空间，运行传统聚类算法在该特征空间上获得图的分割。除此之外，KL 算法[13]也是一种比较具有代表性的算法，该算法是一种试探优化算法，主要应用于网络中节点的分割。算法结果是将社区划分为相等的两个子社区，首先随机产生两个节点集合，分别计算两个集合中每个节点的内部消耗和外部消耗，之后交换两个集合中的节点，再次计算过程中损失的差值，直至损失值为负数算法停止。该算法的精确度不高，有待改进。

潜在的空间模型(Latent space models)算法[14]，是将网络中的节点映射到一个低维度的欧几里得空间内，这样可以在新形成的空间内保持各节点之间的网络连接关系，随后在低维度空间内使用类似于 k-means 的算法对节点进行聚类。其代表算法是 multi-dimensionalscaling(MDS)[15]算法，算法需要输入一个相似性矩阵$P \in R^{n \times n}$，

矩阵中的每个元素P_{ij}代表节点i和节点i在网络中有关联，算法使用的距离度量是常见的测地距离，即两个节点之间的最短路径的长度。此算法可以看作是一个优化问题，寻找I个向量$X_i, \ldots, X_j \in R^N$，使之被嵌在一个子空间R^N，I个向量放在这个子空间中，彼此之间的相似性尽量被保留，求得最优解的过程中可以使用各种数值优化的方法。

Block 块模型算法[16]中一个模块代表一个类，因此我们可以近似地把网络的相互作用描述为$A \approx S \sum S^T$，其中，S是模块指标矩阵，$S \in (0,1)^{n \times k}$，$\sum$ 是块之间的相互作用密度，k为模块数量，此时，算法就转化为优化问题，即求得$\min \|A - S \sum S^T\|_F^2$的最优解。算法优化过程中可以使用各种聚类算法使之达到优化的目的。其代表算法有稠密子图提取方法[17]，该算法通过对相似矩阵的排序以及不断移除图内的边来达到分割的目的。

模块度最大化的社区发现算法[18]解决了上述算法无法确定网络划分的类结构的适宜数量的问题。该算法通过考虑节点的度分布来对真实世界网络的社区划分结果进行测量。算法中，Newman 提出了“模块度”这一概念用以评价网络划分满意度。模块度 Q 是由 Newman 在二十世纪初在一篇论文中提出的(即 FN 算法)[19], 即随机给定一个网络，Q 值用来评估网络内社区划分结果的好坏。它的物理意义是社区内节点之间连接的边的总数与随机情况下的连接边数之差，这类算法的关键在于通过不断优化 Q 值来提高网络社区发现的精度。

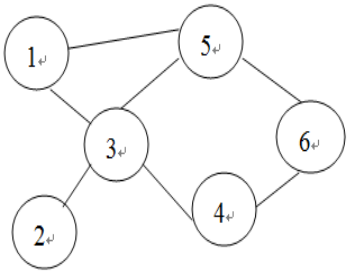
目前，社区发现以获得众多国内外学者的广泛关注，并在社交网络、生物信息和交通运输等领域获得了成功的应用。然而，它仍然面临的众多挑战，如网络数据的特征提取、社区快速发现和重叠社区发现等问题，需要人们继续对其进行深入地研究。

1.3 网络数据及其相关符号的描述

表 1.1 本文涉及的相关符号以及注释

符号	注释
$G = G(V，E)$	网络无向图
$V = \{v_1, v_2, \ldots v_n\}$	网络无向图中节点的集合
$E = \{E_j \mid E_j \in V \times V, j = 1, \ldots m\}$	网络无向图中边的集合
n	网络无向图中节点的个数
m	网络无向图中边的数目
v_i，v_j，v_k	网络无向图中任意一个节点
X	网络原始数据
A	网络无向图的邻接矩阵
d_{min}	无向图中节点之间相似度的阈值
T	$m \times 2$ 的矩阵，E 的数据结构表示
k	全局社区发现算法所发现的类的个数
I	节点对应的与其具有粘合度值的节点序列
d	节点密度
Sim	节点之间的相似度
Coh	粘合度
Rep	节点代表性
$N(v_i)$	节点v_i的邻接点

网络图为$G = G(V, E)$，这里的网络只限于简单网络，即无向（Undirected）、无权（Unweighted）网络，根据节点集V和边集E，可以构造出一个$m \times 2$的矩阵T，由T可得到网络对应的邻接矩阵A。当且仅当节点v_i和节点v_j之间有边时，A_{ij}的值为 1，反之则为 0，A_{ii}统一设为 0。网络的结构如图 1.1 所示。



(a)

1	3
1	5
2	3
3	1
3	2
3	4
3	5
4	3
4	6
5	1
5	3
5	6
6	4
6	5

(b)

$$A=\begin{bmatrix}0&0&1&0&1&0\\0&0&1&0&0&0\\1&1&0&1&1&0\\0&0&1&0&0&1\\1&0&1&0&0&1\\0&0&0&1&1&0\end{bmatrix}$$

(c)

图 1.1 (a)为网络图,(b)为图的矩阵 T,(c)为图的邻接矩阵

1.4 本文的研究内容与组织结构

本文根据国内外社区发现研究现状，在密度聚类技术的支撑下，借鉴已有的社区发现理论与研究成果，对社区发现方法进行了一些深入的研究。主要工作归纳为以下几点：

（1）提出了一种基于密度聚类的全局社区发现算法。该算法定义了新的密度和相似度度量，用来刻画全局社区发现算法中节点对类的代表性以及节点之间的紧密程度，从而判断节点对类的归属。该算法能够快速有效地发现任意形状的社区。

（2）提出了一种基于密度聚类的局部社区发现算法，该算法是在第一个工作的研究基础上，从一个给定节点出发，获得该节点所属的局部社区。该算法不需要获得网络结构的整体信息，大大降低了社区发现的时间复杂度。

全文共分为以下五章：

第一章介绍了社区发现的研究意义以及现状，展示了网络的数据描述以及论文的研究内容和组织结构。

第二章提出了一种基于密度聚类的全局社区发现算法，并通过实验展示了新算法的有效性。

第三章提出了一种基于密度聚类的局部社区发现算法，并通过实验展示了新算法的有效性。

第四章介绍了针对社区发现开发的系统，展示了各种社区发现算法的对比结果，

包括在该系统下各算法的原始数据图，社区发现形状图，生动体现了各算法之间实验结果的差异性，同时具体分析了本文提出的算法与其它对比试验算法在不同评价指标下的不同体现。

第五章对全文进行了总结，指出了本文的创新之处，并对下一步的工作进行了展望。

第二章 基于密度聚类的全局社区发现算法

2.1 密度聚类算法

相比层次聚类算法[20]和划分聚类算法[21]等，密度聚类算法对类的形状有着较强的鲁棒性，能够发现任意形状的类结构。密度聚类算法认为每个类都是由一组高密度的节点组成，被低密度的节点（即噪声）分隔开来。而如何快速对低密度区域的节点进行过滤是其首要任务[22]。

经典的密度聚类算法主要有 DBSCAN 算法[23]（Density-Based Spatial Clustering of Applications with Noise）和 OPTICS 算法[24]。DBSCAN 算法中包含邻域、核心对象、直接密度可达、密度可达等一系列的定义，通过寻找核心点以及该点的所有密度可达对象，形成目标簇。DBSCAN 的缺点是对参数很敏感，参数的些许不同都能引起结果很大的改变，而参数的选择又无规律可循，只能依靠经验来判断。

为了克服 DBSCAN 两个输入参数依赖于手动输入，并且不同的输入参数会得到不同的聚类结果这两个弊端，提出了 OPTICS 算法，OPTICS 算法中存储了每个节点的可达距离以及核心距离，聚类结果并不是一个显示的类划分结果，而是关于样本点的排序表，算法过程中存储两个队列，一个是核心节点以及它的直接可达对象，另一个队列存储样本点的输出次序，通过不断变换核心点，核心点的可达距离内的节点，和可达距离来进行遍历，直到所有节点都被处理完毕为止。尽管对参数没有过分的依赖，OPTICS 算法的计算代价却很大。

Alex Rodriguez[25]等人在 Science 期刊提出了一种新的密度聚类算法“Clustering by fast search and find ofdensity peaks”（CFSFDP），该算法解决了 DBSCAN 算法对参数敏感和均值漂移算法的计算成本高等问题。算法在距离计算上与 K-MEDOIDS 算法相似，都基于节点之间的距离。并且此算法和 DBSCAN，均值漂移算法类似，能发现非球形的聚类并且自动算出正确的聚类个数。而此算法又与均值漂移算法不同，在均值漂移算法中，聚类中心被定义为节点密度的局部最大值，因此需要得到每个节点所在区域的密度最大值，而此算法无需将节点映射到向量空间，并且不需要准确得到每个节点所在区域的密度最大值，计算更加方便。它具有自己的假设理论基础：聚类中心被小于局部密度的邻居节点包围，并且这些邻居节点与其他局部密度较高的节点距离相对来说很远。对于每个节点来说，我们计算两个相关量：局部密度和分离度，局部密度，分离度定义为节点与其他高密度节点之间的距离最小值。分离度异常大并且局部密度很高的节点往往是聚类中心。在聚类中心找到之后，

其余节点对应放置在它的高密度邻居节点周围。

定义 2.1 设节点为i，则它的局部密度为ρ_i，公式如下：

$$\rho_i = \sum_j x(d_{ij} - d_c) \tag{2.1}$$

其中，d_c是一个截止距离，d_{ij}为节点i,j之间的距离。所以ρ_i是其他节点到节点i的距离小于d_c的节点个数。

定义 2.2 设节点为i，它与周围高密度节点之间的距离最小值为δ_i，公式如下：

$$\delta_i = \min_{j:\rho_j>\rho_i}(d_{ij}) \tag{2.2}$$

由此看出，聚类中心通常就是分离度值异常大的节点。

算法 CFSFDP 描述如下：

Input：网络无向图$G = G(V, E)$，d_c

Output：聚类结果Cluster

Step 1：根据定义2.1和定义2.2，计算每个节点的密度和分离度，密度按降序排列，得到密度排序表；

Step 2：画出关于节点密度和分离度关系的图；

Step 3：找到聚类中心，即分离度和密度都异常大的节点；

Step 4：除去聚类中心，剩余节点按照密度排序表的顺序进行遍历，被分配到离它最近的密度较高的邻接点的类中，并标记此邻接点；

Step 5：以此邻接点为起始节点，重复 *Step 4*，直至所有节点都被遍历过为止；

Step 6：结束。

该算法提出的节点之间的分离度度量（共同的节点个数）依赖于距离度量，计算分离度的时间复杂度为$O(n^2)$，代价很大。所以，该算法并不能直接作用于社交网络。除此之外，由于网络不具有天然的特征空间，部分社区发现通过特征映射提取网络数据的特征空间，该操作时间和空间计算成本昂贵，不太适合处理大型网络。因此，需要定义适合于直接处理网络数据的距离和密度度量方法。在此基础之上，提出一种快捷的社区发现算法，使其有利于大型网络的处理。

2.2 基于密度聚类的全局社区发现算法

定义 2.3 设节点为v_i，则它的邻接点为$N(v_i)$，$N(v_i)$满足以下条件：

$$N(v_i) = \{v_j | (v_i, v_j) \in E\} \tag{2.3}$$

定义 2.4 设节点为v_i，那么节点i的密度$d(v_i)$为：

$$d(v_i) = |N(v_i)| \tag{2.4}$$

其中，$0 \le d(v_i) \le n-1$，这里的n指节点总个数，当$d(v_i) = 0$时，则此点表示一个孤立点。节点的密度越大则该点在社区中的信息交互能力和传播信息能力就越强，与社区中的其它点的联系越紧密。

例 2.1 图 1 是著名的空手道（karate）网络，它的数据模型是美国一所大学的某俱乐部成员之间的朋友关系，该俱乐部曾因是否提高收费标准的问题发生争执，导致分为两个小的俱乐部。karate 网络由 34 个节点，78 条边组成。下图中节点之间的边意味着两个节点是朋友关系，带圆圈的节点为一类，用数字表示的节点属于另一类。

如图 1 所示，边表示节点之间的关联，节点$v1$的度指与该点有关联关系的节点个数，即v_3，v_4，v_5，v_6，v_7，v_8，v_{10}，v_{14}，v_{25}，v_{27}，v_{28}，v_{29}，v_{30}，v_{31}，v_{32}，那么节点v_1的度为 15，使用同样方法计算其余所有节点的密度。下图中，$d(v_1) > d(v_{17})$，节点v_1的信息传播能力明显高于节点v_{17}，和周围节点的关联更紧密，它成为社区中心点的概率就越大。

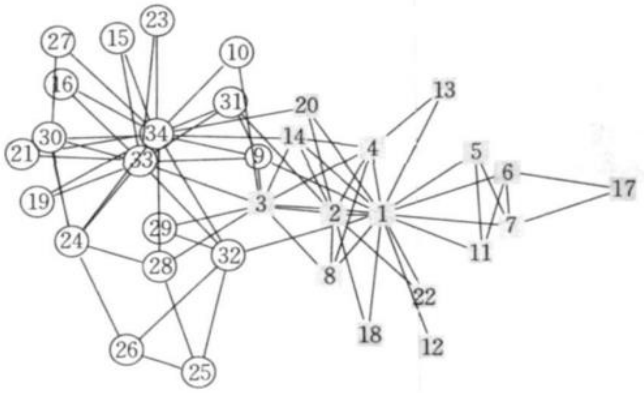


图 2.1 karate 网络无向图

定义 2.5 设节点v_i的密度为$d(v_i)$，节点v_j的密度为$d(v_j)$，两个节点v_i和v_j之间的相似度为$Sim(v_i, v_j)$，公式如下:

$$Sim(v_i, v_j) = |N(v_i) \cap N(v_j)| \tag{2.5}$$

其中，$N(v_i)$和$N(v_j)$分别为节点v_i和节点v_j的邻接点，在聚类过程中，通常使用二者之间的相似性大小或者亲疏远近程度对事物进行划分，相似度高，二者越接近，越可能属于同一类，相似度低，二者关系越疏远。

此处的相似性度量，是对节点之间关系紧密性的刻画和描述，所以需要一种能对节点之间关系进行合理解释的概念，现实中，人与人之间有公共好友这种关系，

即A和B同时与C存在好友关系，那么C就是A，B的公共好友，即$\mathrm{Sim}(A,B)$。于是将节点与节点的相似度定义为：节点与多少个其它节点同时具有联系。这种相似性度量简单方便，被人们广泛使用，为社区发现的顺利进行奠定了坚实的基础。

例 2.2 如图 2.1 所示，节点v_1和节点v_4之间有边，同时节点v_3和节点v_4之间也有边，则说明节点v_4同时和节点v_1和节点v_3有关联关系，那么节点v_4就是二者的公共好友，这样的公共好友有一个相似度就为 1，有两个相似度就为 2，如此类推。

定义 2.6 设Coh代表点的粘合度，如果点v_j密度大于点v_i，且V中点v_j和点v_i相似度最高，那么点v_j和点v_i的相似度值就是我们的粘合度值，公式如下：

$$\mathrm{Coh}(v_j)=\max_{d(v_j)>d(v_i)}\mathrm{Sim}(v_i,v_j) \tag{2.6}$$

某节点与密度比它大的所有节点的相似度序列中，必然存在一个相似度的最大值，它就是所说的节点粘合度值。

例 2.3 如图 2.1 所示，v_1和v_3的公共好友集为$\{v_4,v_8,v_{27}\}$，v_1和v_5的公共好友集为$\{v_7,v_{28}\}$，v_1和v_4的公共好友集为$\{v_2,v_3,v_{10},v_{30}\}$，因为$v_1$和$v_4$公共好友最多，也就是说它们二者之间相似度最高，即粘合度值为 4。计算其他节点的粘合度值时也是如此，逐个比较其公共好友数量。

定义 2.7 设Rep表示某节点对其他节点的代表程度，公式如下：

$$\mathrm{Rep}(v_i)=\frac{d(v_i)}{\mathrm{Coh}(v_i)+1} \tag{2.7}$$

其中，$\mathrm{Coh}(v_i)+1$保证了分母不为 0，出现公式无意义的情况。在复杂网络中，各网络节点都代表独立的个体，每个个体有自己多种多样的特征属性，个体之间的联系有强有弱，个体号召力和在团体内的传播力度也有很大的差异性，因此，在进行社区发现的过程中，由于节点对其他节点的领导能力不同，需要考虑节点的代表性问题，其值由本节点的密度和与本节点的最大粘合度的比值来确定。比值越高，节点的代表性越大，越可能是社区的中心点，比值越低，节点的代表性越弱，越可能是社区的边界节点。

例 2.4 在图 2.1 中，将图 1 中节点按密度大小进行排序，由高到低。计算某点的代表性值时依据以下方法：以节点v_1为例，此点的密度值是 15，粘合度值为 4，节点的代表性：，$\mathrm{Rep}(v_1)=15/4=3.75$，按照此公式依次计算出所有节点的代表性值。一个节点的Rep值越大，表明它对其它节点的代表性越强，根据定义 3.4，节点的代表性值最大者将作为类中心点。

基于密度聚类的全局社区发现算法（Global community discovery algorithm based

on density clustering）描述如下，此算法以下称之为 GCDADC：

Input：原始数据集A，聚类个数k

Output：节点分类集合Cluster，社区中心节点集合Centers

Step1：根据定义 2.3 和定义 2.4，计算社交网络无向图中的所有节点V 的密度，并且从低到高排序，得到密度排序表；

Step2：根据定义 2.5 和定义 2.6，节点与某一节点Sim值最大时，记录此Sim值，除此之外记录对应的节点标号，存入一维数组I中；

Step 3：计算所有节点在社区中的代表性，挑选其中最大的前k个，作为全局社区发现的k个社区中心点；

Step4：根据密度排序表，从第一个社区中心开始，搜索与此中心点有粘合度值的节点，进行标记；

Step5：标记之后的节点为新的起点，重复 *Step 4*，直至搜索到下一社区中心为止，过程中对被标记的节点进行存储；

Step6：以下一个社区的中心为起始节点，重复 *Step4* 和 *Step 5*；

Step7：结束。

算法的总体时间复杂度，关键取决于由邻接矩阵A计算每个节点密度的过程，时间复杂度为$O(n^2)$，计算每个节点与密度大于此节点的节点之间的相似度，时间复杂度最大为$O(n)$，计算每个节点的粘合度所需时间复杂度为$O(n^2)$，其它过程线性时间复杂度几乎可以不计，所以整个算法的时间复杂度主要取决于计算节点密度的操作。

2.3 实验分析

2.3.1 社区发现的评价指标

为了测试算法的性能，选择了在真实世界网络和合成网络两种数据集上进行社区发现实验，并且采用了分别由 Blondel， Martelot 以及 Newman 实现的模块最大化社区发现算法 GCMax1[26]，GCMax2[27]，GCMax3[28]，Danon 提出的贪婪社区发现凝聚算法 GCDanon[29]，J. Hespanha 实现的谱聚类算法 GCSC1[30]，J. Shi and J. Malik 实现的 GCSC2[31]等六种算法作为实验对比算法进行测试。数据集则是从 Network data 中下载的 Word adjacencies，American College football，Political blogs，Books about US politics，Terrist，Web，Word 等数据集，数据集的详细情况见表 2.1。

为了对全局社区发现算法的性能进行评估，在这次试验中我们使用了以下四个有效性评测指标[32]：精度 AC，纯度 PE，调整德兰指标 ARI，标准化互信息 NMI，设$C=\{C_1,C_2,\dots C_k\}$为数据集的一个聚类结果，$P=\{P_1,P_2,\dots P_{k'}\}$是数据集的真实类划分，k 为给定的聚类个数，$n_{ij}$是$C_i,P_j$中共同包含的对象数，$b_i$是$C_i$中的对象，$d_j$是$P_j$中的对象，评测指标被定义如下：

$$AC=\frac{\sum_{i=1}^{k}a_i}{n}$$

$$PR=\frac{\sum_{i=1}^{k}\frac{a_i}{(a_i+b_i)}}{k}$$

$$ARI=\frac{\sum_{ij}\binom{n_{ij}}{2}-\left[\sum_i\binom{b_i}{2}\sum_j\binom{d_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{b_i}{2}+\sum_j\binom{d_j}{2}\right]-\left[\sum_i\binom{b_i}{2}\sum_j\binom{d_j}{2}\right]/\binom{n}{2}}$$

$$NMI=\frac{2\sum_{i=1}^{k}\sum_{j=1}^{k'}\frac{n_{ij}}{n}\log\frac{n_{ij}n}{b_id_j}}{\sum_{i=1}^{k}-\frac{b_i}{n}\log\frac{b_i}{n}+\sum_{j=1}^{k'}-\frac{d_j}{n}\log\frac{d_j}{n}}$$

2.3.2 真实网络实验分析

当社区发现结果越接近真实分类时，AC、PE、ARI、NMI 值越大，随后使用相关算法分别在 Adjnoun，Arm，Football 等真实网络数据集上进行了社区发现结果测试。

表 2.1 数据描述

Data Set	*Nodes*	*Edge*	*Class*
Adjnoun	112	455	2
Football	115	613	12
Polblog	1490	19090	2
Polbooks	105	442	3
Terrist	62	304	5
'Web	75	504	5
Word	112	425	2

表 2.2 到表 2.8 是算法 GCDADC 和其他六种算法在已知社区分类的数据集上的评价指标的比较。

表 2.2 在 Adjnoun 下的算法性能比较

	GCMax1	GCMax2	GCMax3	GCDanon	GCSC1	GCSC2	GCDADC
AC	0.5446	0.5446	0.5268	0.5314	0.5179	**0.7143**	0.5657
PR	0.5490	0.5454	0.5321	0.5314	0.5093	0.7206	**0.8424**
ARI	-0.0117	-0.0130	-0.0138	-0.0180	-0.0011	0.1766	**0.2030**
NMI	0.0051	0.0033	0.0018	0.0065	5.5839e-05	0.1113	**0.1332**
Times	0.3230	0.1580	0.0160	0.2533	1.8454	2.0103	**0.0120**

表 2.3 在 Football 下的算法性能比较

	GCMax1	GCMax2	GCMax3	GCDanon	GCSC1	GCSC2	GCDADC
AC	0.8696	0.8000	0.6348	0.5826	0.8261	0.8522	**0.9130**
PR	0.8877	0.8300	0.7309	0.6530	0.8337	0.8687	**0.9171**
ARI	0.8069	0.7390	0.5364	0.5018	0.7003	0.8037	**0.8493**
NMI	0.8903	0.8626	0.7624	0.7298	0.8161	0.8911	**0.9055**
Times	1.2263	2.5156	1.5262	0.9655	0.7852	0.6659	**0.5896**

表 2.4 在 Polblog 下的算法性能比较

	GCMax1	GCMax2	GCMax3	GCDanon	GCSC1	GCSC2	GCDADC
AC	0.7597	0.7604	0.7597	0.7577	0.5409	0.6329	**0.8349**
PR	0.7972	0.7976	0.7967	0.7981	0.5769	0.7816	**0.9054**
ARI	0.4972	0.5150	0.5207	0.5112	0.0062	0.0702	**0.5448**
NMI	0.3631	0.3725	0.3735	0.3705	0.0123	0.1832	**0.4126**
Times	204.5256	189.8589	156.8747	183.0013	725.2301	536.4452	**137.9040**

表 2.5 在 Polbooks 下的算法性能比较

	GCMax1	GCMax2	GCMax3	GCDanon	GCSC1	GCSC2	GCDADC
AC	0.8381	0.8371	0.8381	0.8371	0.6476	0.8286	**0.8576**
PR	0.6923	0.7217	0.7219	0.7695	0.7427	0.7709	**0.8063**
ARI	0.6280	0.6567	0.6379	0.5580	0.6876	0.5931	**0.8237**
NMI	0.5215	0.5603	0.5308	0.5127	**0.5815**	0.5362	0.5371
Times	0.0780	0.0940	0.0310	0.1252	15.5556	12.2596	**0.0120**

表 2.6 在 Web 下的算法性能比较

	GCMax1	GCMax2	GCMax3	GCDanon	GCSC1	GCSC2	GCDADC
AC	0.5200	0.5200	0.5200	0.5200	0.5200	0.5200	**0.5867**
PR	0.5117	0.5191	0.5028	0.5028	0.5022	0.5152	**0.6709**
ARI	-0.0077	-0.0224	-0.0064	-0.0064	0.0081	-0.0108	**0.3950**
NMI	0.0505	0.0804	0.0766	0.0766	0.0580	0.0723	**0.1484**
Times	0.1560	0.0460	**0.0160**	2.5123	0.5629	1.5624	0.0780

表 2.7 在 Terrist 下的算法性能比较

	GCMax1	GCMax2	GCMax3	GCDanon	GCSC1	GCSC2	GCDADC
AC	0.4516	0.6129	0.4516	0.4516	0.7097	**0.7097**	0.6935
PR	0.7258	0.7370	0.7258	0.7258	0.7182	0.7264	**0.7623**
ARI	0.0812	0.2686	0.0780	0.0780	0.0391	0.0665	**0.2890**
NMI	0.3259	**0.4120**	0.3233	0.3233	0.2832	0.2837	0.2474
Times	0.5631	0.0780	0.0310	12.2565	15.2663	4.2224	**0.0162**

表 2.8 在 Word 下的算法性能比较

	GCMax1	GCMax2	GCMax3	GCDanon	GCSC1	GCSC2	GCDADC
AC	0.2589	0.2589	0.2321	0.2232	0.5179	0.6518	**0.7357**
PR	0.2589	0.2589	0.2324	0.2232	0.5093	0.6513	**0.8424**
ARI	-0.0091	-0.0121	-0.0138	-0.0116	-0.0011	0.0839	**0.3530**
NMI	0.0083	0.0049	0.0018	0.0038	5.5839e-05	**0.0670**	0.0332
Times	0.5225	0.1400	**0.0460**	0.6346	1.5336	1.8632	0.1090

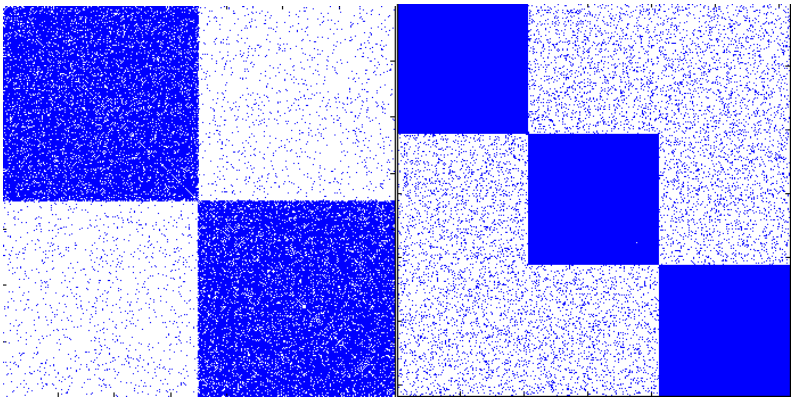
从以上表格中可以看出：总体上，本算法在四个评价指标上均胜过其余六种算法，用于社区发现可以得到相对不错的划分结果。其中，在 Adjnoun 上 GCSC2 算法 AC 值要高于算法 GCDADC，而其余算法在此数据集上表现则较为一致；在 Football 上算法 GCMax3，算法 GCDanon 结果稍微逊色，GCDADC 算法各指标均在优于其他算法；Polblog 中，算法 GCSC1 和算法 GCSC2 的 AC，ARI 数值较低，说明这两个谱聚类方法不适用于发现规模稍大的社区；对于数据集 Polbooks，算法 GCDADC 四个指标表现良好，比另外六种算法指标值高出几个百分点，相对来说更胜一筹。算法在数据集 Web，Terrist，Word 上的结果也类似，算法 GCDADC 在精度，纯度，和 ARI 上都有不错的表现，说明提出的算法 GCDADC 在已知社区结构的网络中，能够精准发现社区结构。为了证明本算法的有效性，在合成网络上也进行了相关实验。

2.3.3 合成网络实验分析

合成网络进行实验时，以两个数据集为实验对象，将第一个数据集分为两个社区，每个社区内有 349 个节点，社区间的边数为 3，社区内部每个节点的边数为 140 条。并将第二个数据集分为 3 个社区，每个社区内有 1028 个节点，社区间的边数为 35，社区内部每个节点的边数为 523 条。

表 2.9 数据描述

Data Set	*Nodes*	*Edge*	*Class*
Data1	698	97720	2
Data2	3084	1612932	3



(d) (e)

图 2.2 (d)Data1 的社区形状图，(e)Data2 的社区形状图

基于这两组数据集，我们分别使用了以上六种算法进行测试，并以 NMI 为评价指标，对算法运行时间进行了记录，实验结果如下表所示。

表 2.10 在 Data1 下的评价指标值比较

	GCMax1	GCMax2	GCMax3	GCDanon	GCSC1	GCSC2	GCDADC
NMI	0.809	0.745	0.781	0.633	0.802	0.768	**0.865**
Times	1.386	0.714	3.502	4.292	0.383	4.693	**0.291**

表 2.11 在 Data2 下的评价指标值比较

	GCMax1	GCMax2	GCMax3	GCDanon	GCSC1	GCSC2	GCDADC
NMI	0.796	0.846	0.882	0.878	0.676	0.768	**0.981**
Times	18.091	11.784	333.605	632.269	9.757	425.042	**6.752**

从以上表格中可以看出：总体上，本算法在评价指标上均胜过其余六个算法，用于社区发现可以得到相对不错的划分结果。其中，在互信息指标值上，算法 GCDADC 对数据集一、数据集二进行的社区发现结果要高于其他算法；在时间复杂度上，对两个数据集进行的社区发现结果中，算法 GCDADC 占优势，这说明本次提出的全局社区发现算法更加快捷高效，在合成网络的虚拟社区结构中，依然能够高效率地发现社区结构。综上所述，算法 GCDADC 在真实网络和合成网络中的社区发现结果都相对稳定，能够有效进行社区发现。

2.4 本章小结

本章讨论了密度聚类的经典算法以及他们的优缺点，提出了基于密度聚类的全局社区发现算法，介绍了新的密度和相似度等度量，并举例进行了说明。实验表明，本算法在真实网络数据和合成网络数据中均表现良好。

第三章 基于密度聚类的局部社区发现算法

3.1 传统局部社区发现算法的局限性

自从社区发现技术提出后，社区发现搜索的对象基本都是整个网络拓扑结构，计算代价大，时间复杂度高。其实，很多情况下，研究关注的对象并不是整个网络的社区结构，也并不是对网络中所有节点都感兴趣，只是需要得到其中一些小团体的相关信息，比如：某个人的高中朋友圈，家庭圈，初中朋友圈等圈子中的一个，这时如果继续对网络中的全部信息进行计算显然并不合适。于是，局部社区发现算法就这样应用而生。

二十世纪初，Fortunato 提出了一种局部社区发现算法 ASJ[33]和一种基于社区适应度[34]的评价方法。该算法开始时将初始社区设为种子节点v_{seed}，通过计算选取使$f(G)$值增幅最大的另一邻居节点加到此社区内部，然后根据社区适应度相关定义重新计算该局部社区内的所有节点的社区适应度，如此反复，当与G节点相邻的所有节点都被访问过，并且再没有能使此局部社区$f(G)$值增大的节点，算法即停止，此时的社区就是节点所在的局部社区。其中，$f(G)$是指G节点的社区适应度，有如下公式：

$$f(G)=\frac{k_{in}^{G}}{(k_{in}^{G}+k_{out}^{G})^{\alpha}} \tag{3.1}$$

其中，k_{in}^{G}是子图的内度和，k_{out}^{G}是子图的外度和，α是控制因子，$f(G)$越大，说明社区越稳定。

同一时期，Papadopolos 通过分析复杂网络的拓扑结构，基于介数中心性产生了边桥度思想，即边桥度的值决定了这条边作为社区之间的桥边的可能性大小。有了边桥度的概念之后，SAA 算法被提出[35]。边桥度公式如下：

$$b_L(e_{st})=1-\frac{|N(s)\cap N(t)|}{\min[(d(s)-1),(d(t)-1)]} \tag{3.2}$$

其中，$N(s)$是节点的邻接点的集合，算法开始时设置某个节点为初始社区，设置了一个阈值作为输入参数，逐步比较该节点与其邻接点的边桥度，边桥度值小于此阈值，则把该节点加入社区，否则舍弃，当社区内没有节点再能加入时，局部社区算法截止，已搜索的节点即构成此局部社区。

这两种算法虽然都不必考虑整个网络拓扑结构，但是选择某节点的邻接点的过程计算成本高，于是，本文提出了一种新的基于密度聚类的局部社区发现算法（Local community discovery algorithm based on density clustering）LCBDC。

3.2 基于密度聚类的局部社区发现算法

定义 3.1 社区中心点：已知节点v_i，当密度大于v_i的节点中不存在相似度大于阈值d_{min}的节点时，节点v_i就是要找的社区中心点。

定义 3.2 社区边界点：已知节点v_j，当密度小于v_j的节点中不存在相似度大于阈值d_{min}的节点时，节点v_j就是要找的社区边界点。社区的边界点到本类中心点距离最远，但是和本类其他节点之间有着密切的关系。

基于密度聚类技术的局部社区发现算法描述如下，以下称此算法为 LCBDC：

Input：原始数据集A，起始节点 n1，节点相似度阈值d_{min}

Output：局部社区集合Cluster，算法运行时间times

Step1: 根据定义 2.3 和定义 2.4，计算社交网络无向图中的所有节点V 的密度，并且从低到高排序，得到密度排序表；

Step2:根据定义 2.5 和定义 2.6，计算无向图中每个节点的粘合度值并记录，除此之外记录对应的节点标号，存入一维数组I中；

Step3:从I中找到与已知节点n1具有粘合度值的对应节点，如果与其粘合度值大于阈值d_{min}，则将此对应节点计入Cluster，反之舍弃；

Step4:设此对应节点为已知节点，重复 *Step3*，直到密度排序表中不再有与已知节点相似度值大于阈值的节点，此时Cluster中的最新节点就是局部社区发现算法的类中心；

Step5:以类中心点为起点逆向搜索密度排序表，重复 *Step3* 与 *Step4*，直到密度排序表中不再有与已知节点粘合度值大于阈值的节点，此时Cluster中的最新节点就是局部社区发现算法的类边界；

Step6:结束。

由于算法 LCBDC 中有对参数d_{min}的选择，在此需要验证此参数的鲁棒性。算法中，相似度阈值d_{min}的选取决定着哪个节点将会成为已知节点的粘合度节点的关键，对算法的精确度和时间复杂度有着至关重要的影响。经过反复试验，得出合适的d_{min}值。下图所示为当原始数据集为 football 时，对d_{min}的选取依据。

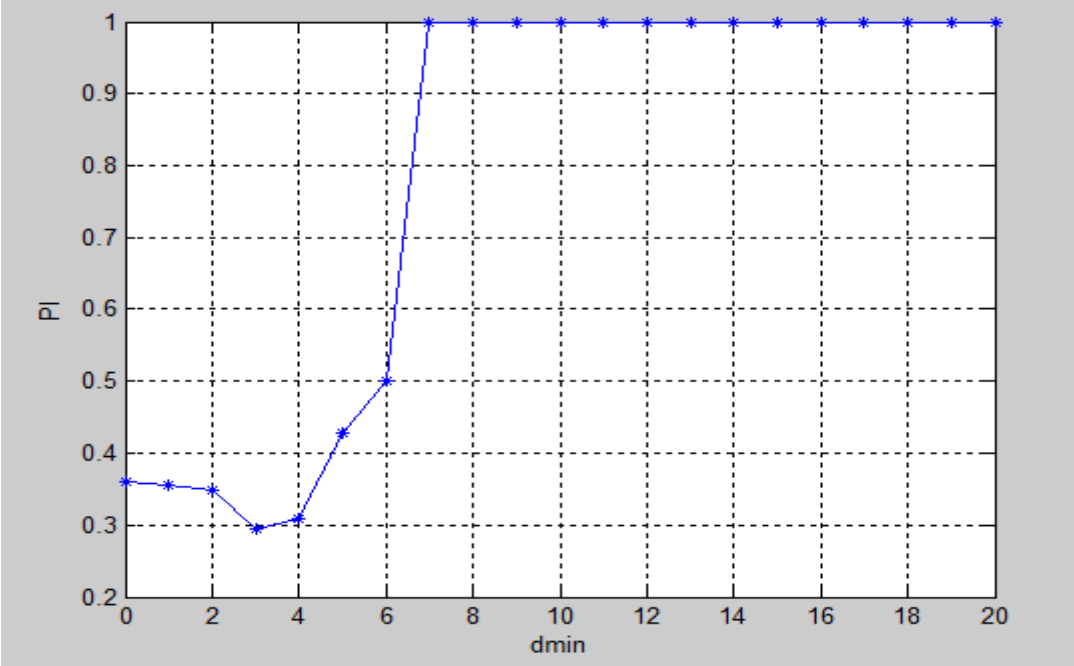


图 3.1 PI 和d_{min}的关系图

例 3.1 如图 3.1 所示，节点集合 $V=\{v_1,v_2,\dots v_8,v_9\}$，边集 $E=\{(v_1,v_2),(v_1,v_3),\dots(v_9,v_7)\}$，$(v_1,v_2)$代表节点$v_1$和节点$v_2$之间有边关联，同理，边集中的其他元素也代表相同含义，接下来，演示根据本算法思想搜索节点 v_5 所属的社区的过程。此例中，$d_{min}=0$。

计算所有节点的密度：$d(v_1)=3;\ d(v_2)=2;\ d(v_3)=3;\ d(v_4)=4;\ d(v_5)=4;\ d(v_6)=4;\ d(v_7)=4;\ d(v_8)=3;\ d(v_9)=1$。

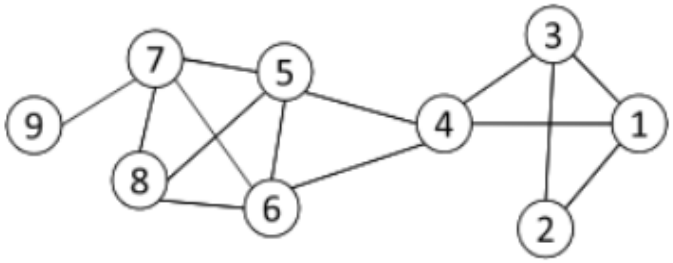


图 3.2 稀疏社区网络无向图

将图中所有按密度大小进行排序，由低到高依次是：$d(v_9)<d(v_2)<d(v_1)<d(v_3)<d(v_8)<d(v_4)<d(v_5)<d(v_6)<d(v_7)$。

其次，计算每个节点的粘合度，这里以节点为例，从密度大于节点v_8的节点中，找出最大相似度值，即粘合度，并记录。$Sim(v_8,v_4)=2;\ Sim(v_8,v_5)=2;\ Sim(v_8,v_6)=2;\ Sim(v_8,v_7)=2$。

综上，取相似度最大值为粘合度值，并且记录节点v_8和节点v_4有粘合度值，计算其他每个节点的粘合度值，都是在密度大于此点的后序排列中找与其有粘合度值的节点。遍历之后，记录每个节点的最大相似度值如下：$\mathrm{Coh}(\mathrm{v}_1)=1;\ \mathrm{Coh}(\mathrm{v}_2)=1;\ \mathrm{Coh}(\mathrm{v}_3)=1;\ \mathrm{Coh}(\mathrm{v}_4)=2;\ \mathrm{Coh}(\mathrm{v}_5)=3;\ \mathrm{Coh}(\mathrm{v}_6)=2;\ \mathrm{Coh}(\mathrm{v}_8)=2;\ \mathrm{Coh}(\mathrm{v}_9)=1$。

除此之外，还需记录每个节点与哪个节点具有最大相似度，由于节点的相似度值都大于$\mathrm{d}_{\min}$，所以v_1的最大粘合度节点是v_3，v_2的最大粘合度节点是v_4，v_3的最大粘合度节点是v_4，v_4的最大粘合度节点是v_7，v_5的最大粘合度节点是v_6，v_6的最大粘合度节点是v_7，v_8的最大粘合度节点是v_4，v_9的最大粘合度节点是v_8。所以根据v_5找到v_6，根据v_6找到v_7，密度排序表中，节点v_7后没有节点，那么v_7就是社区中心点。按照定义 3.2，找到的社区边界点为v_6。据此，将v_5所属的局部社区找出，即$\{\mathrm{v}_5,\mathrm{v}_6,\mathrm{v}_7\}$，在上图中，$\mathrm{v}_5$，$\mathrm{v}_6$，$\mathrm{v}_7$之间联系紧密，因此局部社区发现算法的思想正确，接下来，将在真实网络和合成网络上进行社区发现相关实验，来证明此算法的实用性。

3.3 实验分析

3.3.1 社区发现的评价指标

为了测试算法的性能，依然选择在真实世界网络和合成网络两种数据集上进行社区发现实验，此次采用了最大团算法 MaxClique（Maximum Clique Problem, MCP）[36]作为实验对比算法进行测试，本论文所提出的算法以下称之为 LCBDC。MaxClique 算法思想为：在给定的无向图中找出一个包含节点个数最多的完全子图。完全子图是在具有n个顶点的图中，任意一个顶点均与其余$\mathrm{n}-1$个顶点相邻的图。首先设一个空集合，往这个集合中加入一个顶点，依然构成一个团则加入，否则舍弃。而判断介入此点后是否仍为一个团的标准是此点是否与其他顶点之间都有边。对其它顶点进行递归搜索，直至找到最优解。此问题是一个 NP 完全问题，国际上已开展广泛深入的研究，但国内还处于起步阶段，所以选择 MCP 算法具有现实指导意义。

算法 LCBDC 优于 MCP 算法之处在于：任意给定一个节点 X，都可以通过相关定义找到此点所在的社区，而 MCP 算法是找到网络无向图中存在的最大完全子图，与初始节点无关。因此，算法 LCBDC 能够解决更多关于局部社区发现的问题，

使用更加方便灵活。

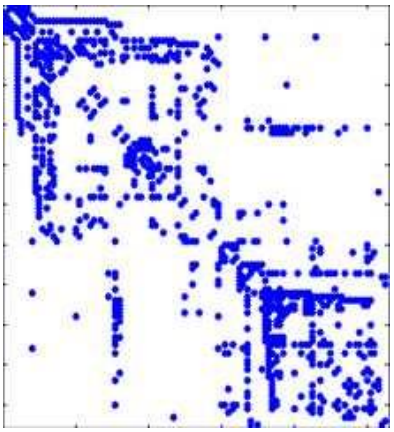
为了对局部社区发现算法的性能进行评估，实验中我们使用了新定义的有效性评测指标：重叠度。设C_i是 LCBDC 算法发现包含给定节点的社区，P_i是数据集中真实包含给定节点的社区。评测指标 PI 公式如下：

$$PI = \frac{|C_i \cap P_i|}{|P_i|} \tag{3.3}$$

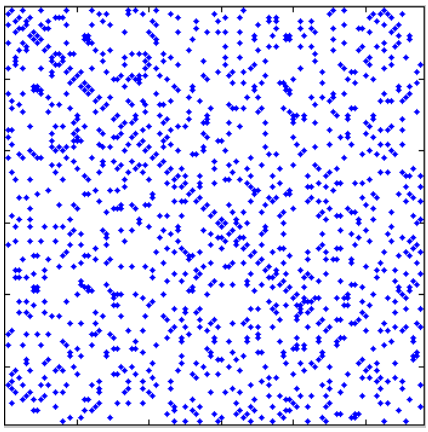
PI 值越高，说明越接近真实分类。

3.3.2 真实网络实验分析

本次实验以 polbooks 和 football 两个真实网络数据集为数据样本，其中，polbook 数据集以 2004 年美国总统大选的政治书籍在亚马逊网上书店销售得出的真实数据为来源，包含了 442 条边和 105 个节点，其中，105 个节点代表 105 本书籍，边则代表两本书曾被同一个购书者购买。数据集 Football 以参加美国 2000 年橄榄球赛季的高校代表队的对战情况为真实数据集，共有 115 个节点，613 条边。数据集如图所示：



(f)



(g)

图 3.3 (f)是 polbooks 的真实网络图，(g)是 football 的真实网络图

算法 LCBDC 在 polbooks 下的社区发现结果如下图，图中紫色点代表算法 LCBDC 发现的一个局部社区，蓝色点代表不属于同一社区的其他节点。

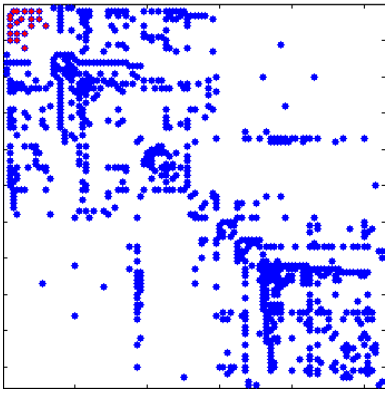


图 3.4 polbooks 下的社区发现结果图

算法 LCBDC 在 football 下的社区发现结果如下图，当给定节点序号为 34 时，节点 34 所在的社区即红色节点所在区域，红色点代表算法 LCBDC 发现的某一局部社区，蓝色点代表不属于此社区的其他节点。

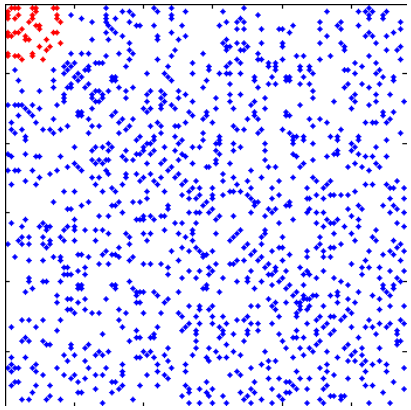


图 3.5 football 下的社区发现结果图

局部社区发现算法 LCBDC 以及最大团算法 MCP 在 polbooks 和 football 下的对比实验数据如下。

表 3.1 在 polbooks 下的算法性能比较

	MCP	LCBDC
PI	0.547	**1**
Time	0.273	**0.004**

表 3.2 在 football 下的算法性能比较

	MCP	LCBDC
PI	0.846	**0.973**
Time	23.319	**0.005**

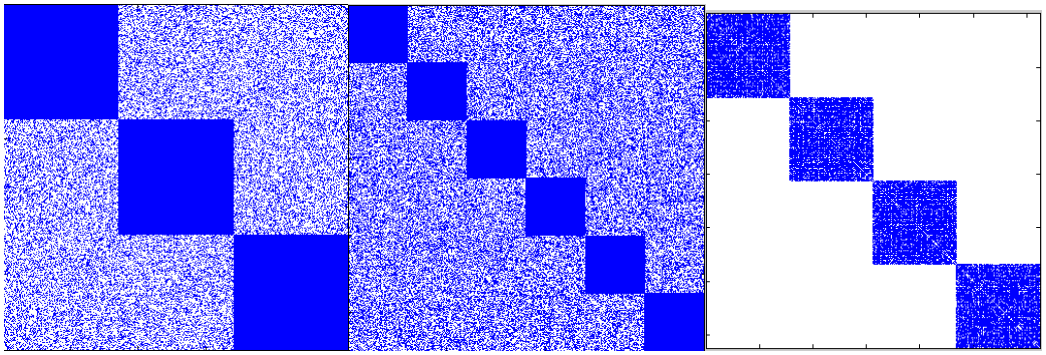
根据以上实验结果，在模拟的合成数据集上，局部社区发现算法 LCBDC 在运行时间和效果上都优于传统的最大团算法 MaxClique，社区发现的结果更加准确，精确度更高。综合来看，效果比较稳定。

3.3.3 合成网络实验分析

对合成网络进行实验时，我们选取了三个数据集，第一个为每个社区内有 852 个节点，共 3 个社区，社区内部每个节点的边数为 400 条。第二个为每个社区内有 300 个节点，社区内部每个节点之间的边数为 156 条，共 6 个社区。第三个数据集为每个社区内节点个数为 156，分为 4 个社区，社区内部每个节点的边数为 80 条。

表 3.3 数据描述

Data Set	*Nodes*	*Edge*	*Class*
Data1	2556	1022400	3
Data2	1800	280800	3
Data3	624	49920	4



(H) (I) (J)

图 3.6 (H)是 Data1 的合成网络社区形状图，(I)是 Data2 的合成网络社区形状图，(J)是 Data3 的合成网络社区形状图

表 3.4 在数据集 Data1 下的算法性能比较

	MCP	LCBDC
PI	0.812	**0.917**
Times	405.185	**40.457**

表 3.5 在数据集 Data2 下的算法性能比较

	MCP	LCBDC
PI	0.68	**0.893**
Times	114.332	**9.773**

表 3.6 在数据集 Data3 下的算法性能比较

	MCP	LCBDC
PI	0.659	**1**
Times	5.862	**0.272**

根据以上实验结果，在所模拟的合成数据集上，局部社区发现算法 LCBDC 在运行时间和效果上同样优于传统算法 MaxClique，并且效果稳定，相较而言，算法 LCBDC 社区发现的结果更加准确，精确度更高。

3.4 本章小结

本章总结了已有的局部社区发现算法的局限性，并在此基础上提出了一种新的局部社区发现算法，并简述了算法原理。同时通过实验表明，该算法相较于最大团算法在时间复杂度以及社区发现精确度上有明显的提高。

第四章 社区发现实验系统的设计与实现

4.1 系统功能

本系统在“社区发现工具箱”(CDTB)[37]的基础上进行了二次开发，主要功能包括：数据导入，算法选择，评价函数选择，社区发现结果显示四个模块。其中，数据导入模块由真实网络数据以及合成网络数据的导入组成；而算法选择模块则支持全局社区发现算法和局部社区发现算法的选择；社区发现结果显示模块包括社区发现图像显示，评价指标值显示，以及运行时间的显示等。功能模块图如图所示。

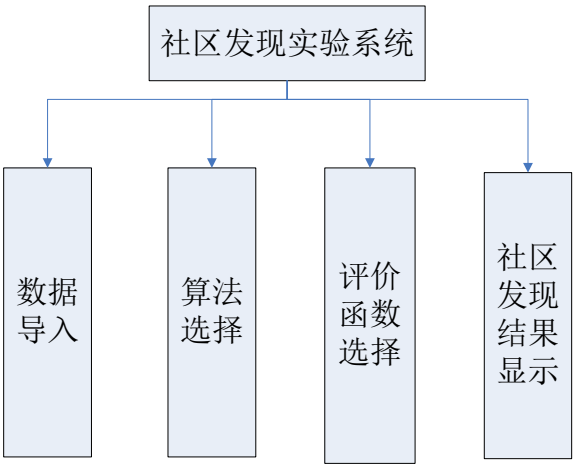


图 4.1 系统功能模块图

4.2 全局社区发现算法的效果演示

选用图生成算法构造具有社区结构的网络，如图 4.2 左下角所示，选定所用算法后，根据界面提示，继续选择社区发现相关算法，评价指标算法，确定之后点击相应按钮，显示相应的社区发现结果以及评价指标的值。

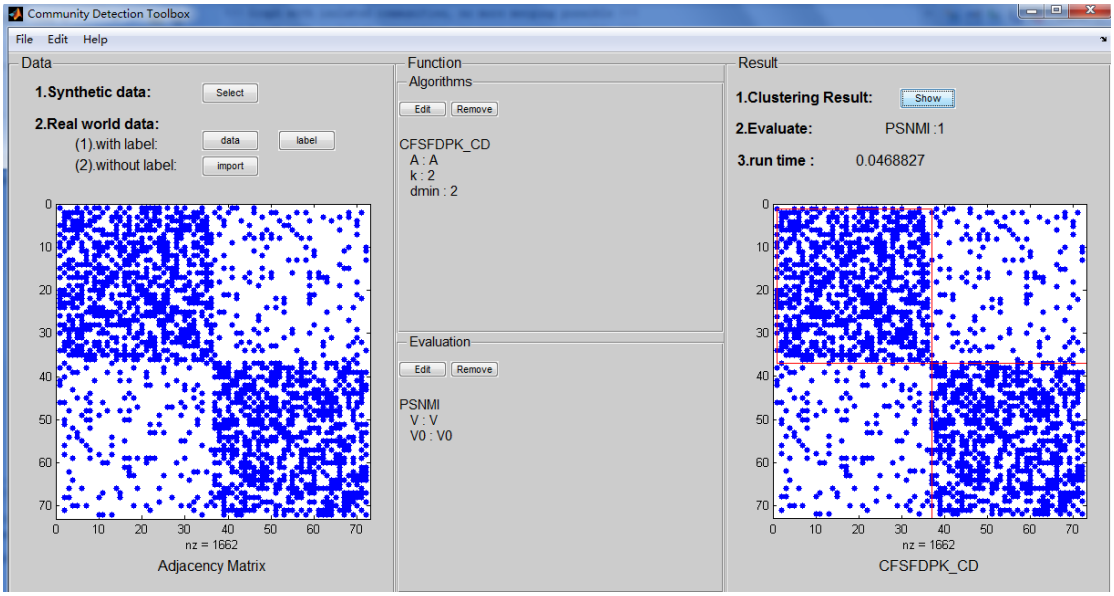


图 4.2 全局发现算法演示图

如图 4.3，生成图的参数输入为$N = 48$，$k = 3$，$Z_i = 25$，$Z_e = 3$，即生成的无向网络图的每个社区有节点 48 个，聚类个数为 3 个，每个社区内的边数为 25，社区之间的边数为 3。本文提出的 GCDADC 全局社区发现算法 NMI 评价指标值为 1，比其余算法 NMI 值都要高，运行时间又较短，总体来说，性能比较稳定。

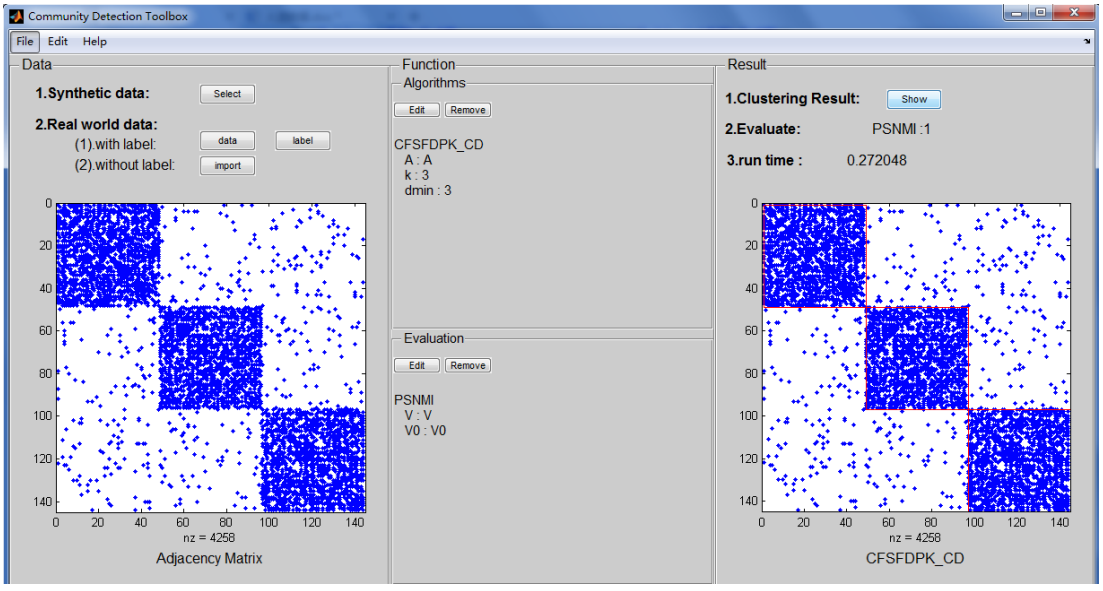


图 4.3 全局社区发现算法演示图

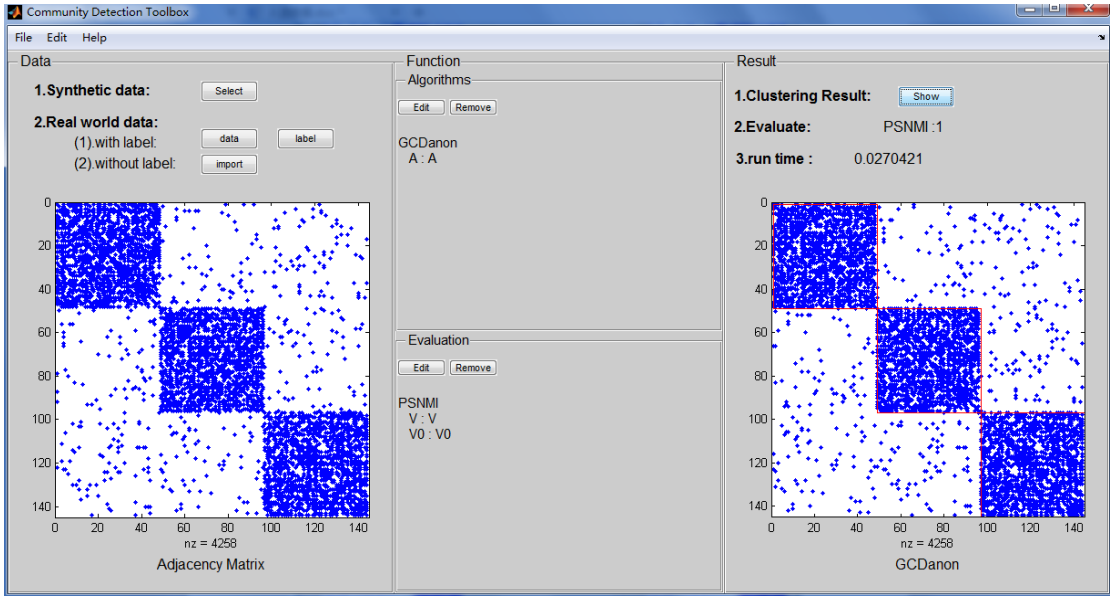


图 4.4 全局社区发现算法演示图

由以上实验数据得知，算法 GCDADC 在准确率上保持得较为稳定，PSNMI 评价指标值不输于任何一个算法的值，算法运行时间上，有时会稍逊于对比算法，接下来，我们还可以对其进一步改进。

4.3 局部社区发现算法的效果演示

LCBDC 在真实网络数据 polbooks 下的实验结果如图 4.5 所示：当给定节点序号为 17 时，通过使用局部社区发现算法 LCBDC，找到图中右下方所示的社区，形状接近于正方形，说明找到的类内部连接紧密，而且相较而言，社区发现所用的时间比 MCP 算法短，NMI 值为 1，算法效率较高。输入其他节点序号时，结果类似于此图，NMI 值为 1，社区发现所用的时间较短。

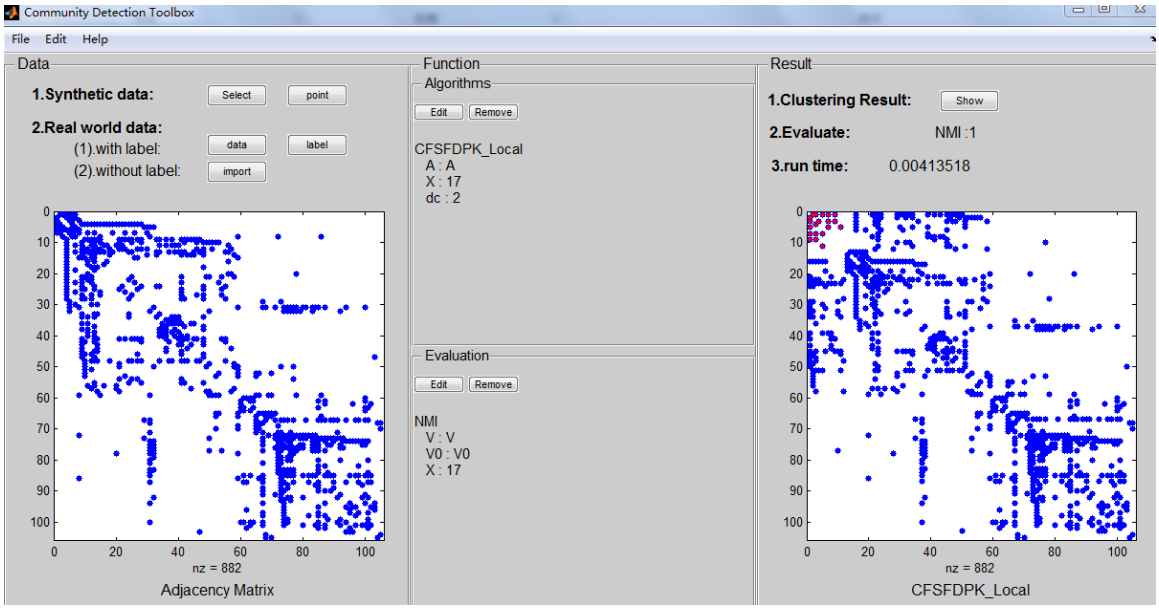


图 4.5 局部社区发现算法在 polbook 下的演示图

算法 LCBDC 在真实网络数据 football 下的实验结果如图 4.6 所示：左下方为 football 中节点的散乱分布，当给定节点序号为 38 时，算法 LCBDC 所发现的社区为图中红色区域，蓝色区域为不属于此社区的节点，评价指标 NMI 值为 0.77778，运行时间短暂，为 0.0102305s。

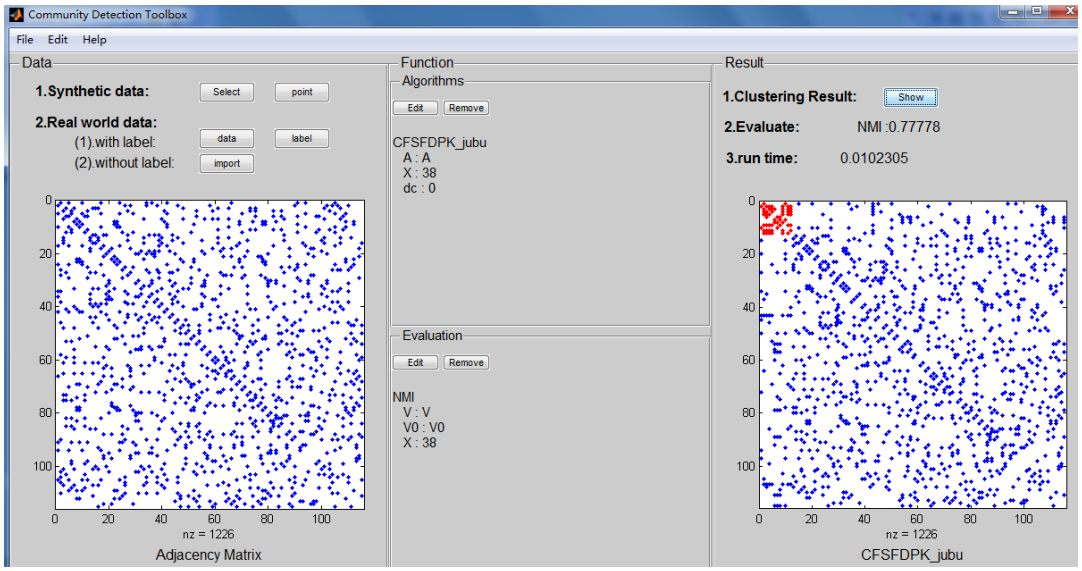


图 4.6 局部社区发现算法在 football 下的演示图

如下图所示，在合成网络数据集中，共两个社区，每个社区内有 36 个节点，每个社区内节点数均等，当以节点 37 为起始节点时，发现局部社区的结果如图所示，红色区域为已知节点 37 所在的社区，即所发现的局部社区，蓝色区域是不属于同一社区的节点，算法运行时间不到一秒，互信息 NMI 值为 1，局部社区发现算

法的结果接近类标签的分类。

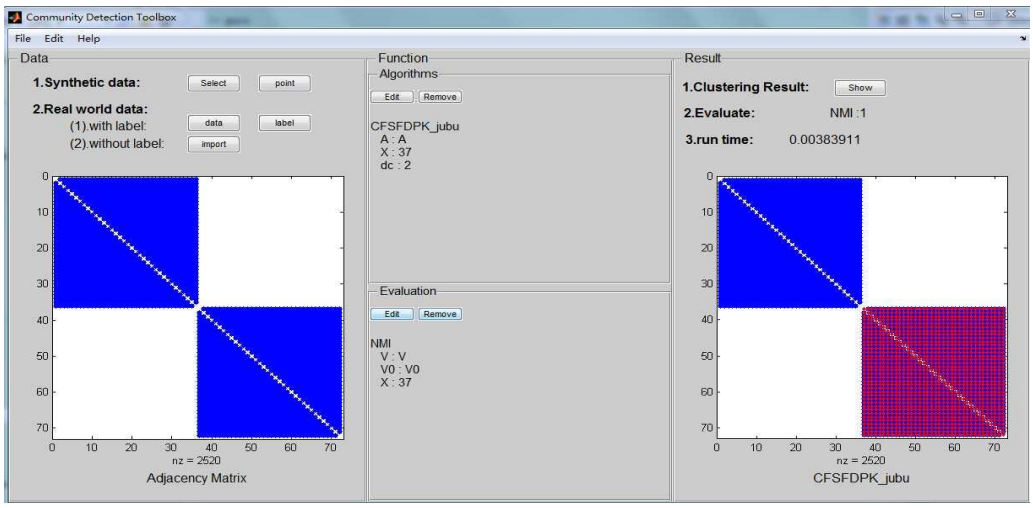


图 4.7 局部社区发现算法合成网络演示图

已知节点为 14 时，即寻找节点 14 所在的社区，由于合成的社区类别按序号排列，按节点序号来说，由于生成网络时每个社区节点个数为 36，节点 14 则应属于第一社区，而红色部分表示已知节点所在的社区，结果相符，并且与原来社区的划分完全相同，互信息 NMI 值为 1。结果如图 4.8 所示：

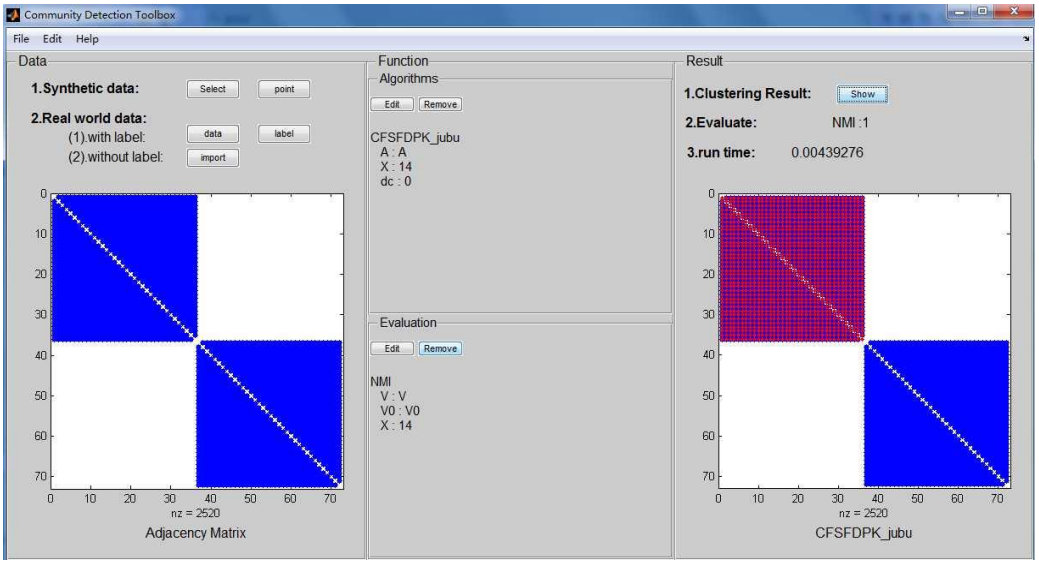


图 4.8 局部社区发现算法合成网络演示图

上述合成网络中，单个社区内的节点个数的数量级基本是个位数，属于小型网络，为了验证此局部社区发现算法在较大的数据集下依然有效，以下将要合成的网络每个社区内节点在10^2以上，每个社区内的边数也在10^2以上，实验结果如下图所示：

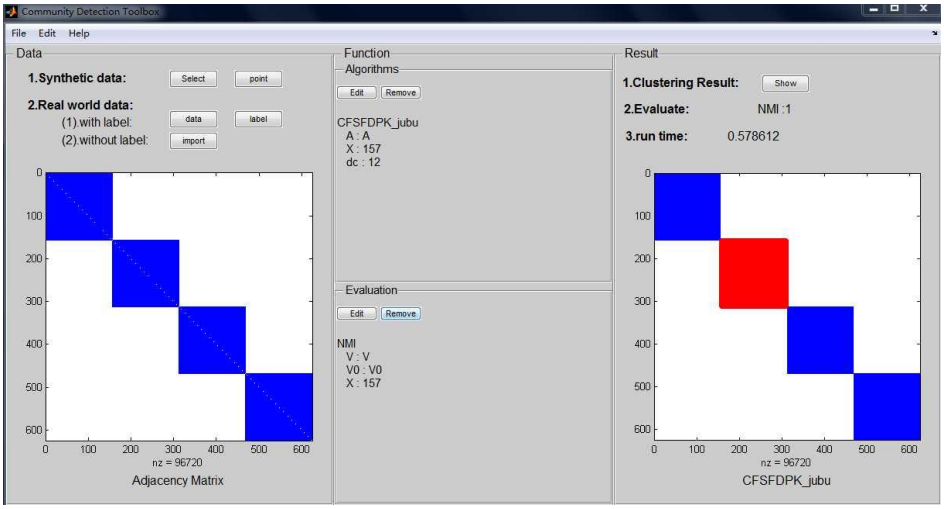


图 4.9 局部社区发现算法合成网络演示图

当输入节点序号不同时所找到的所属社区也不同，节点序号在第一类时，找到的社区在第一个正方形内；节点序号在第二类时，找到的社区在第二个正方形内，以此类推。

4.4 本章小结

本章简要介绍了社区发现算法演示系统所具备的功能，并对各个功能模块的实时效果进行了演示，方便读者了解系统的作用和具体操作流程。

第五章 结论与展望

5.1 总结

本文首先阐述了课题的研究背景和研究意义，以及国内外研究现状，在现有社区发现算法的基础上，结合了密度聚类技术，分别提出了新的全局和局部社区发现算法，并且在真实网络和合成网络的数据集上进行了实验，并对实验结果进行了详细的阐述。从实验结果来看，两种社区发现算法均取得了不错的成果。

论文在社区发现算法中有以下几点创新之处：

1、充分利用密度聚类能发现不同形状社区的特性，将密度聚类算法扩展到了网络社区发现中。由于网络数据不具有特征空间，而传统的密度聚类算法必须先将网络数据映射到特征空间下，在特征空间中计算两两节点之间的距离，整个过程耗时长，计算代价大。本文对此进行了改进，提出了新的密度和相异性度量，将并其直接应用于网络数据，节约了计算成本，提高了时间复杂度，能够更好地服务于网络数据的社区发现。

2、由于很多情况下用户更关注某一类信息，社区发现工作者掌握网络全局信息愈加困难，本文提出了一种局部搜索算法，即通过一个给定的节点，快速锁定所在社区。算法中还提出了新的快速寻找社区中心点和边界点的方法，通过与其它算法的比较表明，该局部社区发现算法在时间复杂度和社区发现精确度上更胜一筹。

5.2 展望

对于社区发现算法的研究，尽管在硕士生研究工作中已取得了一些成果，但限于研究时间，这些成果还需进一步的补充和完善，主要在以下几个方面：

（1）本文提出的两种社区发现算法并没有考虑同一节点可能属于多个社区的情况，但是在许多真实网络中，社区结构具有重叠性，即网络可以划分为多个社区，网络的某个节点可以属于不同的社区。正因为如此，本文的算法在实际应用中可能会有一定的局限性。

（2）复杂网络中个体和个体之间的联系纷繁复杂，这些联系中难免有部分是冗余的，如何快速高效地去除这些冗余边，保留网络内关键信息，减少社区发现工作量，进一步提高社区发现的效率是当下迫切需要解决的问题，这需要我们进一步探索。

参考文献

[1] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms. Phys. Rev. E78, 046110[J]. Physical Review E, 2008, 78(4 Pt 2).

[2] 王林,戴冠中. 复杂网络中的社区发现——理论与应用[J]. 科技导报,2005,08:62-66.

[3] Liu C, Jing L, Jiang Z. A Multiobjective Evolutionary Algorithm Based on Similarity for Community Detection From Signed Social Networks[J]. Cybernetics IEEE Transactions on, 2014, 44(12):2274-2287.

[4] 沈华伟, 程学旗,陈海强,刘悦. 基于信息瓶颈的社区发现[J]. 计算机学报,2008,04:677-686.

[5] Sun P G, Gao L. A framework of mapping undirected to directed graphs for community detection[J]. Information Sciences, 2015, 298:330-343.

[6] Mcauley J, Leskovec J. Discovering Social Circles in Ego Networks[J]. Acm Transactions on Knowledge Discovery from Data, 2012, 8(1):73-100.

[7] Boykov Y, Veksler O, Zabih R. Fast Approximate Energy Minimization via Graph Cuts[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2001, 23(11):1222-1239.

[8] 段志生. 图论与复杂网络[J]. 力学进展, 2008,06:702-712.

[9] Boykov Y, Funka-Lea G. Graph Cuts and Efficient N-D Image Segmentation [J]. International Journal of Computer Vision, 2006, 70(2):109-131.

[10] Michael Wels, Gustavo Carneiro, Alexander Aplas, et al. A Discriminative Model-Constrained Graph Cuts Approach to Fully Automated Pediatric Brain Tumor Segmentation in 3-D MRI[C], 2008:67-75.

[11] Malik J, Shi J. Normalized cuts and image segmentation [J]. IEEE Trans.pattern Anal.mach.intell, 2000, 22(8):888-905.

[12] 秦洋, 王立宏, 武栓虎,等. 基于拉普拉斯矩阵的 DNA 序列集相似性分析[J]. 北京交通大学学报:自然科学版, 2009, 33(6):137-140.

[13] Laporte G. The traveling salesman problem: An overview of exact and approximate algorithms[J]. European Journal of Operational Research, 1992, 59(2):231-247.

[14] Sarkar P, Moore A W. Dynamic Social Network Analysis using Latent Space Models [J]. Sigkdd Explorations Special Issue on Link Mining, 2015, 7(2):31-40.

[15] Shepard R N. The analysis of proximities: Multidimensional scaling with an unknown distance function. I [J]. Psychometrika, 1962, 27(2):125-140.

[16] Olhede S C, Wolfe P J. Network histograms and universality of blockmodel approximation [J]. Proceedings of the National Academy of Sciences of the United States of America, 2013, 111(41):14722-14727.

[17] O'Neil J, Szyld D B. A block ordering method for sparse matrices[J]. Siam Journal on Scientific & Statistical Computing, 1990, 11(5):811-823.

[18] Newman M E. Modularity and community structure in networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2006, 103(103):8577-82.

[19] Newman M E. Fast algorithm for detecting community structure in networks.[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004, 69(6):066133-066133.

[20] Corpet F. Multiple sequence alignment with hierarchical clustering.[J]. Nucleic Acids Research, 1988, 16(22):10881-90.

[21] Park N H, Lee W S. Statistical s-Partition Clustering over Data Streams[J]. Lecture Notes in Computer Science, 2003:387-398.

[22] 周水庚, 周傲英, 金文,等. FDBSCAN:一种快速 DBSCAN 算法[J]. 软件学报, 2000(6):735-744.

[23] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discoverying clusters in large spatial databases with noise[C], 1996:226--231. 2004, 24(4):45-46.

[24] Ankerst M, Breunig M M, Kriegel H P, et al. OPTICS: ordering points to identify the clustering structure[C], SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, Usa. 1999:49--60.

[25] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science,2014,12:344-1492.

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

[27] Martelot E L, Hankin C. Multi-scale Community Detection using Stability as Optimization Criterion in a Greedy Algorithm [C]. Proceedings of the 2011 International Conference on Knowledge Discovery and Information Retrieval (KDIR 2011), 2011: 216-225.

[28] Newman M E. Fast algorithm for detecting community structure in networks.[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004, 69(6):066133-066133.

[29] Danon L, Diazguilera A, Arenas A. Effect of size heterogeneity on community identification in complex networks[J]. Journal of Statistical Mechanics Theory &Experiment, 2006, 2006(11):11010.

[30] Hespanha P. An Efficient MATLAB Algorithm for Graph Partitioning[J]. 2004.

[31] Malik J, Shi J. Normalized cuts and image segmentation[J]. IEEE Trans.pattern Anal.mach.intell, 2000, 22(8):888-905.

[32] Danon L, Díazguilera A, Duch J, et al. Comparing community structure identification[J]. Journal of Statistical Mechanics Theory & Experiment, 2005, 2005(09):09008.

[33] Lancichinetti A, Fortunato S, Kertész J. Detecting the overlapping and hierarchical community structure of complex networks[J]. New Journal of Physics, 2008, 11(3):19-44.

[34] 高启航，景丽萍，于剑,等. 基于结构和适应度的社区发现[J]. 中国科学技术大学学报, 2014(7).

[35] Papadopoulos S, Skusa A, Vakali A, et al. Bridge Bounding: A Local Approach for Efficient Community Discovery in Complex Networks[J]. Psychonomic Science,

2009, 1(1-12):174.

[36] Gregori E, Lenzini L, Mainardi S. Parallel K-Clique Community Detection on Large-Scale Networks[J]. IEEE Transactions on Parallel & Distributed Systems, 2013, 24(8):1651-1660.

[37] Mitalidis M, Kehagias A, Gevezes T, Pitsoulis L. Community Detection Toolbox (CDTB), http://www.mathworks.com.

攻读学位期间取得的研究成果

致谢

本学位论文是在我的导师白亮的悉心关怀下完成的，在书写论文期间，白老师给了我无微不至的关照，在研究方面也给了我很多的指导。他对学术深深的热爱，以及严谨的工作作风深深感染了我，让我明白以怎样的态度面对科研，使我有了长足的提高。在此谨向白老师致以诚挚的谢意和崇高的敬意。

感谢学校和学院的各位师长，谢谢你们曾经有声的无声的教导，让我受益良多。谢谢你们为我提供的学习机会和优越的科研环境，使我能够在良好的学习氛围中工作和学习，感谢同门的师弟师妹们，在科研过程中给与我的鼓励和一些力所能及的帮助，籍此向他们表示诚挚的感谢。

感谢我的家人和朋友，多少年来你们一直默默支持着我，让我不断完成学业，攀登科学的高峰。在这期间，你们的付出我都铭记于心。是你们的关心呵护才让我有勇气和动力在面临任何困难的时候不畏艰难、奋力前行。向所有关心和帮助我的人们表示衷心的感谢！

最后衷心感谢在百忙之中评阅论文和参加答辩的各位！

个人简况及联系方式

个人简况

姓名：赵越

性别：女

籍贯：山西省晋中市

个人简历

2010.9-2014.7 就读于山西省山西大学计算机与信息技术学院，学士

2014.9-至今 就读于山西省山西大学计算机与信息技术学院，攻读硕士学位

联系方式

联系电话：15698407705

电子邮箱：15698407705@163.com

承诺书

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

作者签名：

20　年　月　日

学位论文使用授权声明

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

保密的学位论文在解密后遵守此协议。

作者签名：

导师签名：

年　月　日