基于广义后缀树的二分网络社区挖掘算法

邹凌君[1] 陈 崚[2,3] 戴彩艳[4]

(金陵科技学院信息化建设与管理中心 南京 211169)[1] (扬州大学信息工程学院 扬州 225009)[2]

(南京大学计算机软件新技术国家重点实验室 南京 210093)[3]

(南京航空航天大学计算机科学与技术学院 南京 210016)[4]

摘 要 近年来,二分网络的社区挖掘问题得到了极大的关注。提出了一种基于广义后缀树的二分网络社区挖掘算法。首先从二分网络的邻接矩阵中提取网络中每个节点的链接节点序列,然后构建广义后缀树。广义后缀树的每个节点表示二分网络的一个完全二分团,由此获取并调整完全二分团。通过引入二分团的紧密度得到初始的社区划分,最后再对孤立点进行处理以得到最终的社区划分。所提算法不仅能发现重叠社区,而且能得到一对多关系的社区。在人工数据集和真实数据集上的实验表明,所提算法能准确地识别二分网络中的社区个数,获得很好的划分效果。

关键词 二分网络,社区划分,广义后缀树,重叠社区

中图法分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.07.039

Detecting Community from Bipartite Network Based on Generalized Suffix Tree

ZOU Ling-jun[1] CHEN Ling[2,3] DAI Cai-yan[4]

(Information Technology and Management Center, Jinling Institute of Technology, Nanjing 211169, China)[1]

(College of Information Engineering, Yangzhou University, Yangzhou 225009, China)[2]

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)[3]

(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)[4]

Abstract In recent years, the problem of detecting communities from bipartite network has drawn much attention of researchers. This paper presented an algorithm based on generalized suffix tree for detecting communities from bipartite networks. The algorithm firstly extracts the adjacent node sequence for each node from the adjacency matrix of the bipartite network, and constructs a generalized suffix tree. Each node in the generalized suffix tree represents a complete bipartite clique. Then the algorithm extracts and adjusts those cliques. The closeness of two cliques is introduced to form initial communities. Finally, isolated nodes are processed to get the final community partition. The proposed algorithm can detect overlapping communities, and is able to get one-to-many correspondence between communities. Experimental results on the artificial networks and real-world networks show that, our algorithm can not only accurately identify the number of communities from bipartite networks, but also obtain high quality of community partitioning.

Keywords Bipartite network, Community division, Generalized suffix tree, Overlapping communities

1 概述

自然界和人类社会中的很多复杂系统都可以转化成复杂网络,如生物系统、经济系统、生态系统等。在网络中,节点代表对象,边表示节点之间的连接关系。大部分的网络都具有一定的社区结构。社区结构的概念最早是由 Newman[1] 提出的,即一个复杂网络可以划分成若干个社区,社区内部的节点连接紧密,相互作用较强;社区之间的节点连接稀疏,节点间的相互作用较弱[2]。这些社区结构与网络的功能结构和组织密切相关,通常对应着不同的功能单元。从复杂网络中挖掘社区结构,能帮助人们深入理解网络的拓扑结构,挖掘隐含的信息,预测网络行为。近年来,分析复杂网络的社区结构得到了许多学者的关注,同时出现了很多社区挖掘算法[3-9]。

二分网络是复杂网络的一种重要表现形式,在自然界的网络中具有普遍性。二分网络由两类不同的节点组成,同类节点之间不存在连边。现实世界的科学家-论文网[10]、演员-影视作品网[11]、听众-歌曲网络[12]、疾病-基因的作用网络[13]等都呈现出自然的二分结构。

近年来,对二分网络的社区挖掘是当前复杂网络研究领域的一个研究热点[14-25]。对二分网络的社区挖掘主要有两

到稿日期:2017-01-02 返修日期:2017-03-05 本文受国家自然科学基金项目(61379066),江苏省高校自然科学基金项目(15KJD520008),江苏省现代教育技术研究重点课题(2017-R-54927)资助。

邹凌君(1984—),女,硕士,工程师,主要研究方向为数据挖掘、人工智能,E-mail:njzoulingjun@163.com;**陈 崚**(1951—),男,教授,博士生导师,主要研究方向为数据挖掘、人工智能、并行与分布式处理;**戴彩艳**(1985—),女,博士,主要研究方向为复杂网络链接预测、数据挖掘。

种方法：1)把二分网络映射到单分网络[14-15]，用单分网络的社区发现算法进行研究，但该方法会产生原始信息丢失以及投影后导致单分网络的边剧增等问题；2)直接在原始的二分网络上进行社区发现，这种方法保留了原始的结构和统计特性。为了评估社区挖掘的质量，Newman[16]提出了模块度来对单分网络的社区划分结果进行量化。Guimerà等[17]定义了一种全新的二分模块度，并在此基础上提出了一种社区挖掘算法，但该算法每次只能划分一种类型的节点。Barber[18]对 Newman 的单分模块度进行了拓展，定义了新的二分模块度，提出了 BRIM 算法，该方法需要提前获知社区的数量，且只能发现一对一这种对应关系的社区。Murata[19]认为节点和社区之间不应局限于一对一的关系，从而提出了更加适用于发现一对多社区关系的二分模块度。此外，Murata 等[20]还提出了 LP&BRIM 算法，将 Raghavan 等[3]提出的标号传播方法 LP 和 BRIM 算法结合起来，可在大型二分网络中获得较好的划分质量。Xu 等[21]提出了一种基于蚁群优化的社区发现算法 ACODC，将社区发现问题转化为优化问题，建立了一个可供蚂蚁搜索的图模型。汪涛等[22]提出了一种基于图正则化的三重非负矩阵分解算法以用于二分网络社区发现，该算法能够快速且准确地挖掘二分网络的社区结构。

本文提出了一种基于广义后缀树的二分网络社区发现算法。使用广义后缀树来表示二分网络，通过搜索后缀树，得到完全二分团，然后根据完全二分团的紧密度生成初始社区，最后对孤立点进行处理以得到最终的社区划分。算法无需事先指定社区个数，能发现重叠社区以及一对多关系的社区，且能在有限的时间内获得更好的划分效果。

2 问题定义

二分网络由两类不同的节点和连边组成，同类节点之间不存在连边。如图 1 所示的二分网络中，矩形是同一类型的节点，圆形是另一类型的节点，同一类型的节点之间没有连边，连边只存在于不同类型的节点之间。

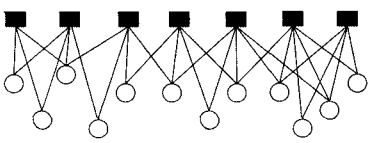

图 1 二分网络的一个示例

二分网络可以用二部图 $G=(V^X,V^Y,E)$ 表示，V^X 和 V^Y 分别是 G 的两个节点集，E 是节点的连边。对于 E 中的边 $(V_i^X,V_j^Y)\in E$，必有 $V_i^X\in V^X,V_j^Y\in V^Y$。假定 V^X 有 n 个节点，V^Y 有 m 个节点，则二部图的邻接矩阵如式(1)所示：

$$\tilde{A}=\begin{bmatrix}0_{n\times n} & A_{n\times m}\\ (A^{\mathrm{T}})_{m\times n} & 0_{m\times m}\end{bmatrix} \tag{1}$$

其中，$0_{n\times n}$ 和 $0_{m\times m}$ 是 $n\times n$ 和 $m\times m$ 的全零矩阵。$A_{n\times m}$ 和 $A_{m\times n}^{\mathrm{T}}$ 是 $n\times m$ 和 $m\times n$ 的非零矩阵。因为邻接矩阵是对称的，所以可以用矩阵 $A_{n\times m}$ 来代表二部图 G，矩阵的每一行代表 U 中一个节点的链接关系，每一列表示 V 中一个节点的链接关系。将 $A_{n\times m}$ 称为二部图的关系矩阵。

二分网络的社区挖掘就是将二部图 $G=(V^X,V^Y,E)$ 划分成 C 个子图，$G_i=(V_i^X,V_i^Y,E_i)$，$i=1,2,\cdots,C$，其中 $V_i^X\in V^X$，$V_i^Y\in V^Y$，$\bigcup_{i=1}^{C}V_i^X=V^X$，$\bigcup_{i=1}^{C}V_i^Y=V^Y$，对于 E_i 的任意边 (u,v)，必有 $u\in V_i^X,v\in V_i^Y$，即每一个子图 G_i 也是一个二部图。社区划分要求每一个子图 G_i 内部节点间的链接尽可能地多，而不同社区之间的节点链接尽可能地少。

为了评价社区划分的质量，Newman 提出了一个衡量单分网络社区划分的指标——模块度。Murata 在文献[19]中分析比较了 Guimerà等人和 Barber 等人提出的适用于二分网络的模块度，并提出了一种能发现社区间一对多关系的二分模块度。设 M 为二分网络的边数，V 是网络中所有的节点数，V_l 和 V_m 是不同类型的节点构成的社区内的节点集合，$A(i,j)$ 是二部图邻接矩阵的值，若节点 i 和节点 j 之间有连边，则 $A(i,j)=1$，否则 $A(i,j)=0$。

首先定义 e_{lm}，其表示节点集 V_l 连接节点集 V_m 的边在总边数中所占的比例。

$$e_{lm}=\frac{1}{2M}\sum_{i\in V_l}\sum_{j\in V_m}A(i,j) \tag{2}$$

其次定义一个 $k\times k$ 的对称矩阵，以 e_{lm} 为矩阵的第 l 行 m 列元素，矩阵第 l 行元素之和 a_l 定义为：

$$a_l=\sum_m e_{lm}=\frac{1}{2M}\sum_{i\in V_l}\sum_{j\in V}A(i,j) \tag{3}$$

在此基础上，Murata 的二分模块度定义如下：

$$Q_B=\sum_l Q_{B_l}=\sum_l(e_{lm}-a_l a_m),m=\arg\max_k(e_{lk}) \tag{4}$$

Q_{B_l} 越大，l 社区和 m 社区的对应关系越强。模块度 Q_B 表示二分网络划分方案的优劣程度，Q_B 越大则表示该划分方案越好。

3 基于广义后缀树的二分网络社区挖掘算法

3.1 后缀树

后缀树是一种广泛应用于字符串索引的数据结构，它可以被应用于多个研究领域，如生物信息学、文本编辑、聚类等。

设有字符集 Σ，$|\Sigma|$ 表示字符集的大小，Σ^* 是使用 Σ 构造的所有可能的字符串。设 $S=s_0s_1s_2\cdots s_{n-1}\$$ 是定义在 Σ 上的长度为 n 的字符串，$\$$ 是终止字符，$\$\notin\Sigma$。$S_i=s_is_{i+1}s_{i+2}\cdots s_{n-1}\$$ 表示 S 的第 i 个后缀字符串。字符串 S 的后缀树表示为 T，存储了 S 的所有后缀。从根节点到叶节点的路径与 S 的后缀一一对应，每条边用一个非空子串表示。后缀树中除根节点外有两类节点，即中间节点和叶节点，中间节点的每条边均被 S 的后缀的一个前缀标记。同一节点发出的两条边的标签必须以不同的字符开始。

目前有很多成功的后缀树构造算法，Ukkonen 算法[26]是应用最广泛的一种构造方法，具有较高的效率，时间复杂度为 $O(n)$。

广义后缀树是存储了一组字符串的所有后缀的后缀树，本文使用广义后缀树进行二分网络社区的挖掘。

3.2 算法的基本思想与框架

本文算法包括 6 个步骤：1)从二部图的关系矩阵中提取节点序列。首先构造二部图 G 的关系矩阵 A，A 的每一行代表一个节点的链接方式。从 A 的每一行中提取该节点的节点序列，在 A 的第 i 行中，值为 1 的元素对应的列下标构成了第 i 个节点的节点序列。2)根据节点序列构建一个广义后缀

树，树中的每一个节点表示 G 中的一个完全二分团。3)从广义后缀树中获取初始完全二分团。4)调整初始完全二分团。5)根据完全二分团的紧密度形成初始社区。6)由于有一些孤立点未被包含在任何初始团中，因此根据孤立点的紧密度将其分派到关系最紧密的社区中，从而形成最终的社区划分。

下面介绍各个步骤的具体细节。

步骤 1　从二部图的关系矩阵中提取节点序列。

定义 1(二部图的节点序列)　设二部图 $G=(V^X,V^Y,E)$ 的关系矩阵为 $A=(a_{ij})$，则二部图中节点 $i(i\in V^X)$ 的节点序列表示为：

$$S_i=(j^1,j^2,\cdots,j^k),a_{ij^k}=1 \tag{5}$$

节点序列表示了两类节点之间的链接关系。图 2 所示的二分网络 $G=(V^X,V^Y,E)$ 由两类节点 V^X 和 V^Y 组成。V^X 有 4 个节点，V^Y 有 5 个节点。$V^X=\{u_1,u_2,u_3,u_4\}$，$V^Y=\{v_1,v_2,v_3,v_4,v_5\}$。$G$ 可以用一个 $4*5$ 的关系矩阵表示。

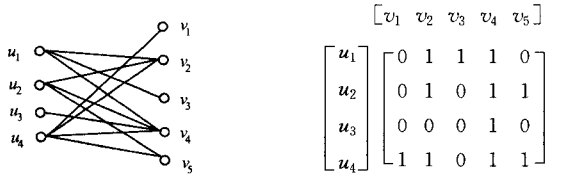

图 2　二分网络及其关系矩阵

节点集 V^X 中的节点序列为：$S_1=(2,3,4)$，$S_2=(2,4,5)$，$S_3=(4)$，$S_4=(1,2,4,5)$。其中 $S_1=(2,3,4)$ 表示 V^X 中 u_1 节点和 V^Y 部分的 v_2,v_3,v_4 之间都有边相连。

步骤 2　根据节点序列构建广义后缀树。

根据步骤 1 得到的节点序列构建广义后缀树。在后缀树中，用一个非空子串标记每条边，并为每个节点存储一个二元组数据(n,B)，其中 n 是整数，记录经过该节点的子序列个数；B 是一个集合，$B=\{p|S_p$ 经过该节点且 p 为整数$\}$。

由节点序列构造的广义后缀树如图 3 所示。

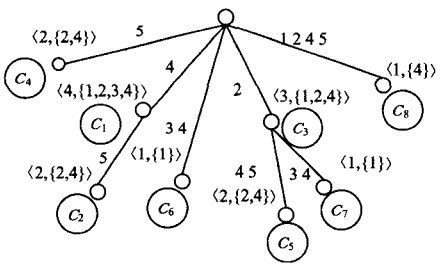

图 3　广义后缀树

步骤 3　从广义后缀树中获取初始完全二分团。

定义 2(完全二分团)　广义后缀树中的每个节点代表网络的一个完全二分团 C_i，表示为 $C_i=(V_i^X,V_i^Y)$，其中 $V_i^X\in V^X$，$V_i^Y\in V^Y$。

V_i^Y 是从根节点到树中这一节点路径上的字符集合，$V_i^X=B$。V_i^X 中的每个节点分别和 V_i^Y 中的节点相互之间有边相连。

从图 3 可得到如下的初始完全二分团：$C_1=(\{1,2,3,4\},\{4\})$，$C_2=(\{2,4\},\{4,5\})$，$C_3=(\{1,2,4\},\{2\})$，$C_4=(\{2,4\},\{5\})$，$C_5=(\{2,4\},\{2,4,5\})$，$C_6=(\{1\},\{3,4\})$，$C_7=(\{1\},\{2,3,4\})$，$C_8=(\{4\},\{1,2,4,5\})$。

步骤 4　调整初始完全二分团。

由步骤 3 产生的初始完全二分团中存在冗余的团，需要将其删除。另外，可通过合并初始完全二分团产生新的团。

定义 3(可合并的完全二分团)　若(I_1,J_1)和(I_2,J_2)是完全二分团，则$(I_1\cap I_2,J_1\cup J_2)$(其中 $I_1\cap I_2\neq\emptyset$)和$(I_1\cup I_2,J_1\cap J_2)$(其中 $J_1\cap J_2\neq\emptyset$)也是完全二分团。

根据定义 3，可将 $C_1=(\{1,2,3,4\},\{4\})$和 $C_3=(\{1,2,4\},\{2\})$合并成一个新的完全二分团 $C_9=(\{1,2,4\},\{2,4\})$。

定义 4(冗余完全二分团)　若 $C_1=(I_1,J_1)$和 $C_2=(I_2,J_2)$是完全二分团，且 $I_1\supseteq I_2$，$J_1\supseteq J_2$，可将其标记为 $C_1\supseteq C_2$，则 C_2 是冗余完全二分团。

本例中，对于团 $C_2=(\{2,4\},\{4,5\})$，$C_5=(\{2,4\},\{2,4,5\})$，因为 $C_2\subseteq C_5$，所以将 C_2 删除。

同样，因为 $C_3\subseteq C_9$，$C_4\subseteq C_5$，$C_6\subseteq C_7$，可将团 C_3，C_4 和 C_6 删除。

经过一系列的合并和删除操作，可得到如下团：$C_1=(\{1,2,3,4\},\{4\})$，$C_5=(\{2,4\},\{2,4,5\})$，$C_7=(\{1\},\{2,3,4\})$，$C_8=(\{4\},\{1,2,4,5\})$，$C_9=(\{1,2,4\},\{2,4\})$。

相应的二分网络如图 4 所示。

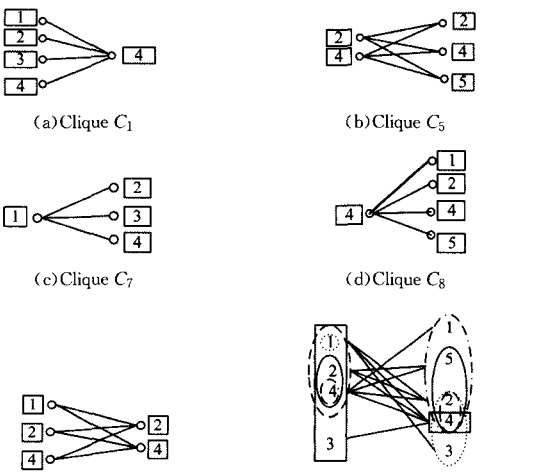

图 4　二分网络中的团

步骤 5　根据完全二分团的紧密度形成初始社区。

从图 4 中可知，不同于社区，完全二分团要求不同类型的节点相互之间都有边连接，而社区只需要节点间紧密连接，且通常有更大的规模。需要对完全二分团进行进一步操作，以得到初始社区划分。

首先删除仅含有 3 个节点的完全二分团，因为这样的团在社区挖掘中没有意义，本例中没有这样的团；其次通过合并联系紧密的完全二分团形成社区，联系紧密的完全二分团是指两个完全二分团有大量重叠的节点和边。

定义 5(完全二分团的紧密度)　设完全二分团 $C_1=(v_1^X,v_1^Y,E_1)$，$C_2=(v_2^X,v_2^Y,E_2)$，v_1^X 和 v_2^X 分别是 C_1 和 C_2 中 X 部分的节点。v_1^Y 和 v_2^Y 分别是 C_1 和 C_2 中 Y 部分的节点。设 $u^X=v_1^X\cap v_2^X$，$u^Y=v_1^Y\cap v_2^Y$ 分别是 $C_i(i=1,2)$中 X 和 Y 重叠的节点，设 $v_i'^X=v_i^X-u^X$，$v_i'^Y=v_i^Y-u^Y(i=1,2)$，则 C_1 和 C_2 的紧密度定义为：

$$R=(C_1,C_2)=\frac{|W(C_1,C_2)|+|Z(C_1,C_2)|}{\min(|W(C_1)|,|W(C_2)|)} \quad (6)$$

其中，$W(C_1,C_2)=\{(i,j)|(i\in v_1'^{x},j\in v_2'^{y})$ or $(i\in v_2'^{x},j\in v_1'^{y}),(i,j)\in E\}$，$Z(C_1,C_2)=\{(i,j)|i\in u^{r},j\in u^{y},(i,j)\in E\}$，$W(C_1)=\{(i,j)|(i,j)\in E_1\}$，$W(C_2)=\{(i,j)|(i,j)\in E_2\}$。

若两个完全二分团的紧密度大于给定的阈值 ε_R，则将其合并成一个社区。本例中，根据式(6)计算 $R(c_1,c_5)=8/4=2$。假定 $\varepsilon_R=1.5$，则将 C_1 和 C_5 合并。

步骤 6 将孤立节点划分到社区中。

经过以上步骤以后，可能仍有一些节点没有被包含在任何一个社区中，需要将这些点划分到与其有最多链接的社区中。

定义 6(孤立点的紧密度) 设 $N(u)=\{v|(u,v)\in E\}$ 是节点 u 的邻居，C 是一个社区，$N(u,C)=\{v|(u,v)\in E,v\in C\}$ 是社区 C 中 u 的邻居集，则节点 u 对社区 C 的紧密度定义为：

$$R(u,C)=\frac{|N(u,C)|}{|N(u)|} \quad (7)$$

设 $C_1,C_2,\cdots,C_m$ 是初始社区，孤立点 u 应被分配到社区 C_k 中，k 满足：

$$k=\arg\max_{1\leqslant i\leqslant m} R(u,C_i) \quad (8)$$

通过式(8)，可以将孤立点分配到相应的社区中。

本文算法描述如算法 1 所示。

算法 1 GSTD(Using Generalized Suffix Tree to Detect Communities)

```
Input:二部图 G=(V^X,V^Y,E)及其关系矩阵 A
Output:G 的社区划分 C1,C2,…,Cm
Begin
  1. For each node i∈V^X do
       计算其节点序列 Si=(j^1,j^2,…,j^k);
     End for
  2. 根据{Si}构建广义后缀树 T;
  3. For 除根节点外,T 中的每个节点 do
       生成完全二分团 Ci=(Vi^X,Vi^Y);
     End for
  4. 通过合并和删除操作调整完全二分团;
  5. For each p∈Ci do
       For each q∈Cj do
         根据式(6)计算 R=(p,q);
           If R>εR then
             合并 p,q,形成初始社区;
           End if
         End for
     End for
  6. For each 孤立点 u do
       据式(8)计算 k=arg max_{1≤i≤m} R(u,Ci);
       将孤立点 u 分配到社区 Ck 中,形成最终的社区划分;
     End for
End
```

3.3 时间复杂度分析

设网络中两部分各有 n 和 m 个节点以及 e 条边，算法的步骤 1 计算节点序列的时间复杂度是 $O(mn)$；步骤 2 将二分网络转换成后缀树的时间复杂度是 $O(e)$；步骤 3 生成完全二分团的时间复杂度是 $O(e)$；由于在合并环节已经事先删除了少于 3 个节点的团，因此合并后每个社区的个数最少是 4 个节点，所以社区的个数要远小于 n 和 m，而且随着合并过程的进行，社区的个数会越来越少，因此步骤 4—步骤 6 中社区的合并、删除和调整过程的时间复杂度为 $O(mn)$。因为 $e<mn$，所以算法的总时间复杂度为 $O(mn)$。

4 实验结果与分析

为了测试本文算法的性能，将算法在人工二分网络数据集以及真实数据集 Southern Women 上进行验证。实验环境为：8GB 内存，Intel core i7 CPU，Windows7 操作系统，使用 Java 编程实现。

4.1 人工二分网络

人工二分网是由计算机生成的 4 类人工网络 $N1-N4$，分别包含 128～1024 个节点。初始社区个数设定为 4，每个社区含有相同的节点个数。二分网络中每个节点的度 D 满足 $D=D_{in}+D_{out}$，其中 D_{in} 表示该节点与本社区内节点的连接数，D_{out} 表示该节点与其他社区内节点连接的个数。显然，D_{out} 越大，社区结构越模糊。人工数据集的参数如表 1 所列。

表 1 人工数据集

	节点数	初始社区个数	初始社区内节点个数	节点度
$N1$	128	4	(16,16)	≤16
$N2$	256	4	(32,32)	≤16
$N3$	512	4	(64,64)	≤32
$N4$	1024	4	(128,128)	≤32

算法设定阈值 $\varepsilon_R=0.4$，分别在各类二分网络上运行算法 10 次并求取平均准确率，结果如表 2 所列。

表 2 不同 D_{out} 值下 4 类人工二分网络的计算准确率的对比/%

数据集 \ D_{out}	1	2	3	4	5	6	7	8
$N1$	100	100	100	100	96.9	90	60.2	45.2
$N2$	100	100	100	97.5	90	80	40.5	30.5
$N3$	100	100	95.5	91.8	86	60	35	25.2
$N4$	100	100	92.5	90.5	81.5	58.5	32	22.5

通过分析表 2 可知，随着社区间连边的增多，社区结构会越来越模糊，因而准确率会有所下降。由此可见，在社区结构明显的情况下，即 $D_{out}\leqslant 5$ 时，算法有较高的准确率，能较好地识别出社团个数和社区结构。

将本文的算法和基于蚁群优化的算法 ACODC[21] 以及基于矩阵分解的 MP 算法[23] 在数据集 $N1$ 上进行对比分析，得到的平均准确率如图 5 所示。

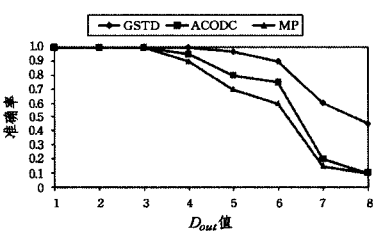

图 5 不同 D_{out} 值下 3 种算法的社区划分平均准确率比较

由图 5 可知，在社区结构明显的情况下，3 种算法都表现

出良好的社区发现性能；随着社区间连边的增多，当 $D_{out}>6$ 时，3 种算法的准确率均呈现下降趋势，但本文算法仍保持了相对较高的准确率，优于其他两种算法，说明本文算法进行社区发现的质量最优。

文中还比较了 3 种算法在不同数据集上的运行时间。分别在各类二分网络上运行算法 10 次，并求得不同 D_{out} 值下的平均运行时间，结果如表 3 所列。

表 3　不同 D_{out} 值下 3 种算法的运行时间比较/s

算法	N1		N2		N3		N4	
	$D_{out}=1$	$D_{out}=7$	$D_{out}=1$	$D_{out}=7$	$D_{out}=1$	$D_{out}=7$	$D_{out}=1$	$D_{out}=7$
GSTD	0.291	0.292	1.97	1.97	12.21	12.23	22.21	22.23
ACODC	0.318	0.320	2.32	2.38	14.01	14.02	26.01	26.02
MP	1.2	1.6	9.8	12.7	23.3	30.2	44.5	56.2

由表 3 可知，3 种算法的运行时间随着数据集的增大而增加。本文算法在不同数据集上的运行时间最短，MP 算法运行时间最长。随着 D_{out} 值的增大，社区结构逐渐模糊，GSTD 的运行时间较为稳定，而 MP 算法的运行时间随着社区结构的减弱呈较大的增幅，这是因为当社区结构较弱时，MP 算法需要反复迭代，从而消耗了大量的运行时间。因此，相比于其他两种算法，本文的算法有更高的时间效率和更好的适应性。

4.2　Southern Women 数据集

使用由 Davis 在 20 世纪 30 年代收集的 Southern Women 数据集作为真实数据集，该数据集描述了密西西比州南方女子俱乐部中 18 名妇女参与 14 项活动的情况。这个数据集有明显的社区结构，所以被广泛地用来测试分析和比较。Southern Women 网络可用一个二部图来表示。该二部图的一部分节点表示妇女，另一部分节点表示活动。如果一个妇女参加了一个活动，那么在该妇女和这个活动的节点之间有一条连边。妇女节点之间以及活动节点之间没有连边。如图 6 所示，对 18 名妇女和 14 个活动进行编号，1－18 号圆点代表妇女，19－32 号矩形点代表活动。

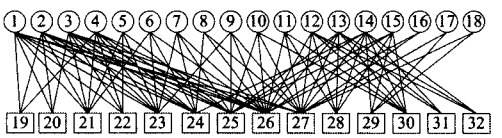

图 6　South Women 二分网络

设置 $\varepsilon_R=0.3$，算法将网络划分为两个社区，即{妇女 1－9，活动 19－27}，{妇女 10－18，活动 25－32}，如图 7 所示。其中活动节点{25，26，27}是重叠节点，本文用 Mutara 模块度来评估该划分的质量，结果是 0.6345。

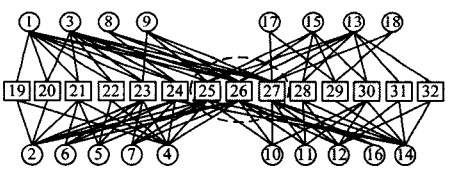

图 7　本文算法在 Southern Women 数据集上的划分结果

表 4 列出了本文算法 GSTD 与 BRIM 算法、LP&BRIM 算法、ACODC 算法、MP 算法的比较情况。

表 4　不同算法对 Southern Women 数据集的实验结果比较

	Women's network	Event network	Murata modularity
BRIM	{1－6}{7,9,10} {8,16－18}{11－15}	{19－24}{25－26} {27,29}{28,30－32}	0.4750
LP&BRIM	{1－7,9}{8,10－18}	{19－25}{26－32}	0.5776
ACODC	{1－9}{10－18}	{19－26}{27－32}	0.5859
MP	{1－6}{7－10} {11－18}	{1－6}{7,8}{9－14}	0.5109
GSTD	{1－9}{10－18}	{19－27}{25－32}	0.6345

由表 4 可知，BRIM 算法将该二分网络分成 4 个社区：{妇女 1－6，活动 19－24}，{妇女 7，9，10，活动 25－26}，{妇女 8，16－18，活动 27，29}，{妇女 11－15，活动 28，30－32}。BRIM 的划分结果经过了 500000 次随机分配的尝试，且需要事先指定社区个数，模块度是 0.4750。Murata 提出的 LP&BRIM 算法将该数据集分成两个社区：{妇女 1－7，9，活动 19－25} 和 {妇女 8，10－18，活动 26－32}，模块度是 0.5776。ACODC 算法将该数据集分为两个社区：{妇女 1－9，活动 19－26} 和 {妇女 10－18，活动 27－32}，模块度是 0.5859。MP 算法将数据集分为 3 个社区：{妇女 1－6，活动 1－6}，{妇女 7－10，活动 7－8} 和 {妇女 11－18，活动 9－14}，模块度是 0.5109。BRIM 算法的模块度最低，本文算法 GSTD 的模块度最高。

本文算法对妇女的划分结果与 ACODC 算法的划分结果一致，与 LP&BRIM 算法的划分结果相差一个节点。Davis 根据人种学知识将妇女划分为{妇女 1－9} 和 {妇女 9－19} 两个社区，第 9 个妇女是两个社区的重叠。为了方便比较，将第 9 个妇女和妇女 1－8 归为一个社区，命名为“Davis1”，将第 9 个妇女和妇女 10－19 归为一个社区，命名为“Davis2”。本文算法对妇女的划分结果与 Davis 提出的一种划分结果一致，算法将活动节点{25，26，27}划分为重叠节点也更加合理。

经过分析可知，与其他二分网络社区发现算法相比，本文算法有许多优点。首先，本文算法不需要事先指定社区个数，目前很多社区发现算法需要事先指定社区个数，这在实际应用中并不可行。其次，本文算法能发现重叠社区，因为初始社区的获取是通过挖掘关系矩阵中所有元素值为 1 的块，而值为 1 的元素可能会被包含在多个这样的块中，这些节点会属于多个社区。最后，本文算法能发现一对多关系的社区。另外，算法具有较低的时间复杂度。

结束语　二分网络的社区发现是当前的研究热点。本文提出了一种基于广义后缀树的社区发现算法，通过遍历后缀树，产生初始的完全二分团作为初始社区，接着根据规则调整初始社区，得到最终的社区划分。本文算法不但能发现重叠社区，同时还能得到一对多关系的社区划分，具有一定的现实意义。在人工数据集和真实数据集上的实验表明，本文算法是有效的，能获得较好的划分结果。下一步的研究重点是当社区结构不明显时，如何更加有效地识别社区结构。

参 考 文 献

[1]　NEWMAN M E J. The Structure and Function of Complex Networks[J]. Siam Review, 2003, 45(2): 167-256.

[2] ADAMIC L A, ADAR E. Friends and neighbors on the Web[J]. Social Networks, 2003, 25(3): 211-230.

[3] RAGHAVAN U N, ALBERT R, KUMARA S. Near linear time algorithm to detect community structures in large-scale networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2007, 76(3 pt 2): 036106.

[4] LIU D Y, JIN D, HE D X, et al. Community Mining in Complex Networks[J]. Journal of Computer Research and Development, 2013, 50(10): 2140-2154. (in Chinese)
刘大有, 金弟, 何东晓, 等. 复杂网络社区挖掘综述[J]. 计算机研究与发展, 2013, 50(10): 2140-2154.

[5] HUANG F L, ZHANG S C, ZHU X F. Discovering Network Community Based on Multi-Objective Optimization[J]. Journal of Software, 2013, 24(9): 2062-2077. (in Chinese)
黄发良, 张师超, 朱晓峰. 基于多目标优化的网络社区发现方法[J]. 软件学报, 2013, 24(9): 2062-2077.

[6] YU H, ZHAO Y L, CUI K, et al. Community Detection Algorithm Based on Cross-Entropy Method[J]. Chinese Journal of Computers, 2015(8): 1574-1581. (in Chinese)
于海, 赵玉丽, 崔坤, 等. 一种基于交叉熵的社区发现算法[J]. 计算机学报, 2015(8): 1574-1581.

[7] JIANG S Y, YANG B H, WANG L X. An Adaptive Dynamic Community Detection Algorithm Based on Incremental Spectral Clustering[J]. Acta Automatica Sinica, 2015, 41(12): 2017-2025. (in Chinese)
蒋盛益, 杨博泓, 王连喜. 一种基于增量式谱聚类的动态社区自适应发现算法[J]. 自动化学报, 2015, 41(12): 2017-2025.

[8] LIU S C, ZHU F X, GAN L. A label-propagation-probability-based algorithm for overlapping community detection[J]. Chinese Journal of Computers, 2016, 39(4): 717-729. (in Chinese)
刘世超, 朱福喜, 甘琳. 基于标签传播概率的重叠社区发现算法[J]. 计算机学报, 2016, 39(4): 717-729.

[9] XIN Y, YANG J, XIE Z Q. Link-Block Method for the Semantic Overlapping Community Detection[J]. Journal of Software, 2016, 27(2): 363-380. (in Chinese)
辛宇, 杨静, 谢志强. 一种面向语义重叠社区发现的 Link-Block 算法[J]. 软件学报, 2016, 27(2): 363-380.

[10] NEWMAN M E J. Scientific collaboration networks. I. Network construction and fundamental results[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2001, 64(2): 016131.

[11] LIU A F, FU C H, ZHANG Z P, et al. An empirical statistical investigation on Chinese mainland movie network[J]. Complex Systems and Complexity Science, 2007, 4(3): 10-16. (in Chinese)
刘爱芬, 付春花, 张增平, 等. 中国大陆电影网络的实证统计研究[J]. 复杂系统与复杂性科学, 2007, 4(3): 10-16.

[12] LAMBIOTTE R, AUSLOOS M. Uncovering collective listening habits and music genres in bipartite networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2005, 72(2): 066107.

[13] CHEN W Q, LU J A, LIANG J. Research in disease-gene network based on bipartite network projection[J]. Complex System and Complexity Science, 2009, 6(1): 13-19. (in Chinese)
陈文琴, 陆君安, 梁佳. 疾病基因网络的二分图投影分析[J]. 复杂系统与复杂性科学, 2009, 6(1): 13-19.

[14] MUKHERJEE A, CHOUDHURY M, GANGULY N. Understanding how both the partitions of a bipartite network affect its one-mode projection[J]. Physica A Statistical Mechanics & Its Applications, 2011, 390(20): 3602-3607.

[15] HORVAT E Á, ZWEIG K A. A fixed degree sequence model for the one-mode projection of multiplex bipartite graphs[J]. Social Network Analysis and Mining, 2013, 3(4): 1209-1224.

[16] 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(23): 8577-8582.

[17] GUIMERA R, SALESPARDO M, AMARAL L A. Module identification in bipartite and directed networks. [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2007, 76(3): 036102.

[18] BARBER M J. Modularity and community detection in bipartite networks[J]. Physical Review E, 2007, 76(6): 066102.

[19] MURATA T. Detecting Communities from Bipartite Networks Based on Bipartite Modularities[C]// Proceedings of the 2009 International Conference on Computational Science and Engineering(CSE'09). Piscataway, NJ, USA: IEEE, 2009: 50-57.

[20] LIU X, MURATA T. Community detection in large-scale bipartite networks[C]// Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology(WI-IAT'09). Washington, DC, USA: IEEE Computer Society, 2009: 50-57.

[21] XU Y C, CHEN L. Community Detection on Bipartite Networks Based on Ant Colony Optimization[J]. Journal of Frontiers of Computer Science and Technology, 2014, 8(3): 296-304. (in Chinese)
徐永成, 陈崚. 基于蚁群优化的二分网络社区挖掘[J]. 计算机科学与探索, 2014, 8(3): 296-304.

[22] WANG T, LIU Y, XI Y Y. Identifying Community in Bipartite Networks Using Graph Regularized-based on-negative Matrix Factorization[J]. Journal of Electronics & Information Technology, 2015, 37(9): 2238-2245. (in Chinese)
汪涛, 刘阳, 席耀一. 基于图正则化非负矩阵分解的二分网络社区发现算法[J]. 电子与信息学报, 2015, 37(9): 2238-2245.

[23] CHEN B L, CHEN L, ZOU S R, et al. Detecting community structure in bipartite networks based on matrix factorization [J]. Computer Science, 2014, 41(2): 55-58, 101. (in Chinese)
陈伯伦, 陈崚, 邹盛荣, 等. 基于矩阵分解的二分网络社区挖掘算法[J]. 计算机科学, 2014, 41(2): 55-58, 101.

[24] CUI Y Z, WANG X Y. Uncovering overlapping community structures by the key bi-community and intimate degree in bipartite networks[J]. Physica A: Statistical Mechanics and Its Applications, 2014, 407(407): 7-14.

[25] BECKETT S J. Improved community detection in weighted bipartite networks[J]. Royal Society Open Science, 2016, 3(1): 140536.

[26] UKKONEN E. On-line construction of suffix trees[J]. Algorithmica, 1995, 14(3): 249-260.