动态社会网络的社团结构检测与分析

刘 瑶，王瑞锦，刘 峤，秦志光

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】真实社会网络如邮件、科学合作、对等网络等均可以用图进行建模。近年来，基于图的社团挖掘吸引了人们越来越多的研究兴趣，它不仅可以帮助识别网络的整体结构，还可以发现社团演变的隐藏规律。尽管使用静态图进行社团挖掘已经被广泛采用，但基于动态图的研究还比较少。通过使用时间序列，对动态图上的社团挖掘包括社团检测与分析进行研究，提出了一个新的动态社团结构检测模型，并采用真实网络数据集进行了实验。实验结果显示该模型在社团结构发现的有效性和效率性方面均有着良好的表现。

关 键 词 社团检测; 动态图; 动态社会网络; 模块度; 时间序列
中图分类号 TP301 文献标志码 A doi:10.3969/j.issn.1001-0548.2014.05.016

Community Detecting and Analyzing in Dynamic Social Networks

LIU Yao, WANG Rui-jin, LIU Qiao, and QIN Zhi-guang

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 610054)

Abstract Real networks such as e-mail, co-author and peer-to-peer networks can be modeled as graphs. Community mining on graphs has attracted more and more attentions in recent years. It not only can help to identify the overall structures of networks, but also can help to discover the latent rules of community evolution. Community mining on dynamic graphs has not been studied thoroughly, although that on static graphs has been exploited extensively. Based on time-sequence, the community mining including community detection and analysis on dynamic graphs is researched in this paper. And a two-step model is presented to discover the dynamic community structure. The effectiveness and efficiency of the model are validated by experiments on real networks. Results show that the model has a good trade-off between the effectiveness and efficiency in discovering communities.

Key words community detecting; dynamic graphs; dynamic social networks; modularity; time sequence

在现实生活中已经越来越离不开各种社会网络，包括社交网站(social network site，SNS)、即时聊天网络、P2P、电子邮件、博客、微博等。社会网络分析也吸引了多个领域研究人员的关注，如现代社会学、人类学、社会语言学、地理、社会心理学、计算机学、信息学、组织研究、经济学及生物学等。

社会网络有很多基本特性：小世界现象、幂律分布、社团结构、重尾分布等。社团结构是社会网络的一个重要特性是指在网络中存在成组的节点，组内的节点相互连接紧密，而组与组之间的节点连接稀疏，每个组就是一个社团。网络的功能是网络的各个社团之间综合作用的结果，知道网络的总体功能并不一定能知道每个社团各自的功能，因此社团结构检测对于理解网络的结构和功能特性具有重要的现实意义：能够更好地理解网络的结构，能够更清晰地认识网络中不同社团之间的关系；每个社团的功能可以由该社团内部的个体的功能来推断；一个社团的成员功能可以由其他成员的功能来推断。

1 动态社团检测算法的相关研究

动态网络中，社团数目可能增加也可能减少，节点的社团归属也可能发生改变。对动态复杂网络进行社团检测，需要考虑网络在不同时刻的演化关系，保证相邻两个时刻的社团划分具有连贯性。

针对某一时刻的静态网络，研究人员已经设计出数量众多的高效社团结构检测算法。而动态网络的社团结构检测研究，由于演化的复杂性和实验数据的匮乏，还处于刚刚起步的阶段。目前，动态复

收稿日期：2013 – 03 – 20；修回日期：2014 – 03 – 19
基金项目：国家高技术研究发展计划主题项目(2011AA010706)；国家自然科学基金(61133016)；中央高校基本科研业务费专项资金(ZYGX2012J067)
作者简介：刘瑶(1978 –)，女，博士生，主要从事网络分析，信息安全方面的研究.

杂网络的社团结构检测主要有以下两类方法：一类是演变社团检测；另一类是通过识别社团演变的关键事件来发现社团的演变模式。文献[1]提出一个新的聚类概念，称为演变聚类，该方法可以捕获社团的演变过程。基于文献[1]提出的时间平滑框架，出现了一些演变聚类算法。文献[2]将动态网络社团检测问题变化为图着色问题，提出了一个对不同时刻网络中的社团进行贪婪匹配的启发式算法。文献[3]提出Facenet算法进行社团检测和演变分析，通过嵌入由历史社团结构得到的时间平滑框架，在一个统一的过程里发现社团并得到其演变趋势。这些已有的方法在实际应用中存在的问题是：假定随着时间的变化社团的数目是固定不变的，而在现实的网络中，社团结构随着时间的演变也会发生相应的变化。即当前的社团数目可能与之前的不同，可能会有新社团的生成、老社团的分解及合并等现象出现。文献[4]提出了一个基于微粒和密度的演变聚类算法，该算法可以发现可变数目的、任意形成和分解的社团，但只能得到社团演变的单一路径，无法识别社团的分解及合并。这种与时间相关的社团检测算法可以得到一组时间平滑的社团序列，因此适用于随着时间推移社团结构比较稳定的网络，社团短时间内不会发生明显的改变。

事实上社团的演变可能有多条路径，为了跟踪社团的演变轨迹，人们致力于标识社团演变的特征事件，如文献[5]提出基于派系过滤的扩展算法来识别社团演变中的关键事件，对连续时刻的网络进行社团检测；文献[6]提出基于匹配的社团事件识别方法，对节点的变化进行分析，对不同时刻的社团关系矩阵使用位操作计算。由于只标识动态网络中社团演变的特征事件，这种分析对大型复杂网络是不实际的。

为了解决上述问题，本文提出了一个基于时间序列的动态社会网络社团结构检测模型。该模型将社会网络的时序动态性和时刻静态性用时间序列的方式表示，在时刻上运行静态社团结构检测算法LMA；然后在时序上运行动态社团结构检测算法DNCD；最后得到社团结构的动态演变轨迹。该模型适用于大规模复杂社会网络。

2　动态社会网络社团检测模型

本文设计的动态网络社团结构检测模型主要包含LMA和DNCD两个算法，通过结构相似度计算模块度，从而对动态网络进行基于时间序列的社团结构检测。该模型分为以下两个步骤。

1) 对 $t=1$ 时刻的原始网络 G_1 运行 LMA 算法，进行静态网络社团结构检测，得到初始的中间过程社团集合 MS_1。

2) 对 $t>1$ 时刻的网络 G_t，DNCD算法将通过LMA算法得到的当前时刻中间过程社团集合 MS_t，与前一时刻的社团时间序列集合进行匹配，发现社团结构的改变；然后，从动态社会网络中检测出稳定的社团集合，并跟踪其动态演变轨迹。

2.1　模块度

文献[7]在解决社团问题时引入了模块度的概念，即在给定社团成员和每个节点度值的条件下，社团内部边与网络总边数之比减去随机网络社团内部边与总边数之比的期望值。它用于判断社团划分的好坏，模块度的取值范围是[−1, 1]，度值越大表明社团划分越合理。在实际应用中，模块度的取值一般在0.3～0.7之间。

在有向图中，模块度为：

$$Q=\frac{1}{m}\sum_{ij}\left[A_{ij}-\frac{k_i^{\mathrm{in}}k_j^{\mathrm{out}}}{m}\right]\delta(c_i,c_j)$$
$$m=\frac{1}{2}\sum_{ij}A_{ij} \tag{1}$$

式中，当 $u=v$，$\delta(u,v)=1$，否则值为0；$\boldsymbol{A}_{ij}$ 是一个邻接矩阵，如果节点 i 和 j 之间有边连接则值为1，否则为0。节点 i 的度为：

$$k_i=\sum_j A_{ij}$$

为了简化描述，定义如下两个变量：

$$e_{uv}=\frac{1}{m}\sum_{ij}A_{ij}\delta(c_i,u)\delta(c_j,v)$$
$$a_v=\frac{1}{m}\sum_i k_i\delta(c_i,v)$$

因为 $\delta(c_i,c_j)=\sum_v\delta(c_i,v)\delta(c_j,v)$，所以模块度的计算可以简化为：

$$Q=\frac{1}{m}\sum_{ij}\left[A_{ij}-\frac{k_ik_j}{m}\right]\sum_v\delta(c_i,v)\delta(c_j,v)=$$
$$\sum_v\left[\frac{1}{m}\sum_{ij}A_{ij}\delta(c_i,v)\delta(c_j,v)-\frac{1}{m}\sum_i k_i\delta(c_i,v)\frac{1}{m}\sum_j k_j\delta(c_j,v)\right]=\sum_v(e_{vv}-a_v^2) \tag{2}$$

2.2 LMA算法

LMA算法使用文献[8]提出的基于相似度的模块度函数，网络中社团内部节点对的相似度要高于分属于不同社团的节点对的相似度。对任意给定网络 $G_t(V_t,E_t)$ 进行社团检测，得到社团集合 $\mathrm{MS}_t=\{C_{t1},C_{t2},\cdots,C_{tn}\}$。

基于相似度的模块度计算：

$$\begin{cases} Q^s=\sum_{i=1}^{NC}\left[\dfrac{\mathrm{IS}_i}{\mathrm{TS}}-\left(\dfrac{\mathrm{DS}_i}{\mathrm{TS}}\right)^2\right] \\ \mathrm{IS}_i=\sum\limits_{u,v\in C_{ti}}\sigma\langle u,v\rangle \\ \mathrm{DS}_i=\sum\limits_{u\in C_{ti},v\in V_t}\sigma\langle u,v\rangle \\ \mathrm{TS}=\sum\limits_{u,v\in V_t}\sigma\langle u,v\rangle \end{cases} \tag{3}$$

式中，NC 是网络中社团的数目；IS_i 是社团 C_i 中节点对的相似度之和；DS_i 是社团 C_i 内的节点与网络中任意节点的相似度之和；TS是网络中所有节点对的相似度之和。基于结构相似度的模块度可以用来衡量社团结构检测的质量，模块度的最大值对应社团结构的最优划分。

为了提高算法效率，只考虑有边连接的社团合并对模块度的影响。对两个社团 C_{ti} 和 C_{tj} 模块度增量 ΔQ^s 的计算为：

$$\Delta Q^s=Q^{C_{ti}\cup C_{tj}}-Q^{C_{ti}}-Q^{C_{tj}}=\frac{2\mathrm{US}_{ij}}{\mathrm{TS}}-\frac{2\mathrm{DS}_{ti}\times\mathrm{DS}_{tj}}{(\mathrm{TS})^2} \tag{4}$$

$$\mathrm{US}_{ij}=\sum_{u\in C_{ti},v\in C_{tj}}\sigma\langle u,v\rangle$$

式中，US_{ij} 是社团 C_{ti} 中节点与社团 C_{tj} 中节点的相似度之和。基于相似度的模块度可以用于评估社团划分的质量，LMA算法用模块度增量控制小社团的合并，如果合并产生的模块度增量 ΔQ^s 大于零，那么这些小社团合并为一个社团。

LMA算法的思想是：给定网络 $G_t(V_t,E_t)$，从网络中的任意一个节点 u 开始，寻找包含 u 的稠密节点对。如果存在节点 v，满足 $\mathrm{TP}(u,v)$ (即 v 与 u 有边连接并且 v 是 u 的邻居集合 $N(u)$ 中与 u 最相似的节点)，那么计算 u 和 v 合并的模块度增量 ΔQ^s；如果 $\Delta Q^s>0$，合并 u 和 v 成为一个小社团。将新生成的小社团看成是节点 u'，再对 u' 寻找其稠密节点对，并计算 ΔQ^s 值，如果 ΔQ^s 值大于零则合并包含 u' 的稠密节点对。重复上述步骤，直到不存在包含该节点的稠密节点对。继续对 V_t 中的下一个未被访问过的节点实施上述步骤，直到网络中的所有节点都被划分成社团。

算法1：LMA算法

输入：$G_t(V_t,E_t)$

输出：MS_t，node，acnode

算法的主要过程：

初始化 $\mathrm{MS}_t=\{u_1,u_2,\cdots,u_n\},Q^s=0$。

1) 从 V_t 中任意选择一个未被访问的节点 u。

2) 对节点 u，找到其邻居节点集合 $N(u)$。

3) 从 $N(u)$ 中查找 $\mathrm{TP}(u,v)$。

4) 如果存在 $TP(u,v)$，并且 $\Delta Q^s>0$，则 $u'=\mathrm{merge}(u,v)$，$\mathrm{MS}_t=(\mathrm{MS}_t-\{u\}-\{v\})\bigcup u'$，$Q^s=Q^s+\Delta Q^s$，$u=u'$，跳转至步骤 3)。

5) 如果 V_t 中存在未被访问过的节点，则转至步骤 2)。

6) 对 MS_t 中的任意社团，如果存在 $|C_{ti}|=1$，则 $\mathrm{MS}_t=\mathrm{MS}_t-C_{ti}$。

7) 对 C_{ti} 中的节点 w，如果存在 $x,y\in N(w)$，$C_{tj},C_{tk}\in\mathrm{MS}_t$，并且 $x\in C_{tj}\bigcap y\in C_{tk}\bigcap C_{tj}\neq C_{tk}$ 则 $w\in\mathrm{node}$；否则 $w\in\mathrm{acnode}$。

8) 算法结束。

2.3 社团演变事件

为了发现社团的演变模式，定义了演变事件来刻画不同时刻的社团演变特征。检测社团演变事件的关键在于如何匹配不同时刻的中间过程社团。文献[13-14]分别定义了刻画动态社区演变特征的事件，但均是针对相邻时刻社团集合的匹配。本文通过计算某一时刻的中间过程社团与之前时刻(不必是连续的时刻)的时间序列社团链关部的相似度来判定社团是否匹配。详细定义如下。

1) 社团的出生：如果在 p 时刻，存在一个中间过程社团 C_{pi}，与 q 时刻($q<p$)的任意社团时间序列链 TSC_k.head 不匹配，那么生成一个新的包含 C_{pi} 的 TSC_m。

2) 社团的死亡：如果在 $p+d$ 时刻，对任意的 TSC_k，至少连续 d 个时刻都没有中间过程社团与之匹配的，那么从 TS 中删掉 TSC_k。

3) 社团的合并：如果在 p 时刻，$\mathrm{TSC.head}=\{\mathrm{TSC}_1.\mathrm{head},\mathrm{TSC}_2.\mathrm{head},\cdots,\mathrm{TSC}_n.\mathrm{head}\}$ 中的每一个链头部均与 q 时刻($q>p$)的一个中间过程社团 C_{qk} 匹配，并且 C_{qk} 与 TSC.head 的节点相同率至少大于 e%，那么将 TS 中的多个链头部合并。

4) 社团的分裂：若在 p 时刻，TSC_k.head 与 q 时刻($q>p$)生成的一个 $\mathrm{MS}_q=\{C_{q1},C_{q2},\cdots,C_{qn}\}$ 匹配，并且 TSC_k.head 与 MS_q 的节点相同率至少大于 e%，那么 TSC_k.head 将分裂。

5) 社团的生长：如果在 p 时刻，存在一个中间

过程社团 C_{qk} 与 q 时刻($q<p$)的某个 TSC_k.head 匹配，且 C_{qk} 大于 TSC_k.head (约10%)，那么 TSC_k.head 将生长。

6) 社团的收缩：如果在 p 时刻，存在一个中间过程社团 C_{qk} 与 q 时刻($q<p$)的某个 TSC_k.head 匹配，且 C_{qk} 小于 TSC_k.head (约10%)，那么 TSC_k.head 将缩小。

2.4 DNCD算法

DNCD 算法通过将不同时刻的社团集合进行匹配来发现社团演变事件，从而获取社团的动态演变轨迹。Hungarian 方法将匹配问题建模成加权偶图匹配问题，然而偶图匹配方法往往假定两个集合中的节点存在 0:1 或 1:1 映射，只能 1 对 1 匹配，无法完全识别社团合并和分解事件[9]。本文算法采用相似度来匹配不同时刻的社团集合，阈值的引入使得多对多匹配成为可能。

算法2：DNCD算法

输入：动态社会网络 $G=\{G_1,G_2,\cdots,G_t\}$，相似度阈值 ε，社团消失时间间隔 d

输出：社团时间序列集合 $\mathrm{TS}=\{\mathrm{TSC}_1,\mathrm{TSC}_2,\cdots,\mathrm{TSC}_n\}$

算法的主要过程：

1) 初始化：使用 LMA 算法从 G_1 中抽取出 $\mathrm{MS}_1=\{C_{11},C_{12},\cdots,C_{1k}\}$，其中 k 表示 MS_1 中包含的中间过程社团数目；将 MS_1 中的中间过程社团依次存入 TS 中，即 $\mathrm{TSC}_1=\{C_{11}\},\mathrm{TSC}_2=\{C_{12}\},\mathrm{TSC}_n=\{C_{1k}\}$。

2) 针对下一个 G_p，$P>1$，抽取出 $\mathrm{MS}_p=\{C_{p1},C_{p2},\cdots,C_{pi}\}$。

3) 针对所有的 $C_{p1},C_{p2},\cdots,C_{pi}$，执行以下步骤：

① 计算 $C_{p1},C_{p2},\cdots,C_{pi}$ 与 TS 中所有社团时间序列链头部的相似度，找出其中相似度大于 ε 的社团时间序列链；

② 如果 $C_{p1},C_{p2},\cdots,C_{pi}$ 中存在与 TS 中的社团时间序列链头部相似度不大于 ε 的中间过程社团，则创建新的社团时间序列链来包含它们；

③ 否则，将相似的中间过程社团加入到其对应的社团时间序列链中。

4) 将 TS 中所有的 TSC.head 指向新加入的中间过程社团。如果某个 TSC.head 与 N 个中间过程社团均相似，则创建 $N-1$ 个新的社团时间序列链来一一包含 $N-1$ 个新的中间过程社团。

5) 重复步骤 2)～步骤 4)，直到处理完所有时刻的网络。

6) 算法结束。

通过匹配 MS_p 和 TS 中所有社团时间序列链的头部，得到相应的社团演变事件。若 C_{pi} 只和一个头部匹配，得到一个生长或收缩事件；若和多个头部匹配，得到一个合并事件；若基于阈值 ε 没有和任何一个头部匹配，则得到一个出生事件。在任意时刻，TS 中的 $\mathrm{TSC}_k=\{C_{ai},\cdots,C_{bj},\cdots,C_{dh}\}$，$a,b,d\in T$ 且 $a<b<d$ 对应一组节点集合 $\{C_{ai}\cup\cdots C_{bi}\cup\cdots\cup C_{dh}\}$，这就是一个有重叠节点的社团集合。根据设置的时间间隔 d，删掉只在很少时刻存在的不稳定的节点集合，得到在多个时刻都存在的稳定社团集合。

3 实验结果与分析

3.1 实验数据集

本文使用真实网络数据集进行实验，以验证算法的有效性和效率。

数据集1：Zachary空手道俱乐部网络

数据集是Wayne Zachary从1970～1972年观察美国一所大学空手道俱乐部34名成员之间的社会关系，构造出的成员关系网络。节点表示成员，节点间的连接代表两个成员经常一起出现在俱乐部活动之处的其他场合。因俱乐部主管John A.(节点34)与教练Mr. Hi(节点1)之间的争执而分裂成2个各自为核心的小俱乐部。网络共包含34个节点，78条边，如图1所示。

数据集2：邮件网络

数据集来自某高校校园网邮件服务器2011年1月1日～12月31日一年的邮件日志记录。考虑到互联网上存在大量的垃圾邮件，会影响人们社会行为模式的数据挖掘效果，邮件服务器上只会完整记录本域邮件用户的活动，而对外部邮件服务器用户的行为记录不完全。为保证数据集的完整性和可靠性，只提取本地邮件用户的日志记录。本地邮件用户的数目在很长一段时期内比较稳定，因此这一数据集具有相对稳定的社团集合。考虑到用户的隐私，使用数字来代替邮件用户地址。

邮件数据集包含5 435个本地邮件用户地址和1 400 740封邮件，每个用户在2011年全年平均收发257.73封邮件。由于系统管理员经常群发邮件给所有本地邮件用户，干扰了正常的行为模式数据挖掘，所以在数据集中去掉包含系统管理员账号的日志记录，如“admin”“webmaster”“mail-Daemon”和“emdg-daemon”；同时也剔除用户自己发给自己的邮件日志记录，即在有向图中一些自循环的边。将邮件数据集映射成如图2所示的有向图，每个节点代

表一个本地邮件用户，每条边代表用户*A*向用户*B*发送一封邮件。共计4 368个活动节点，77 936条有向边。

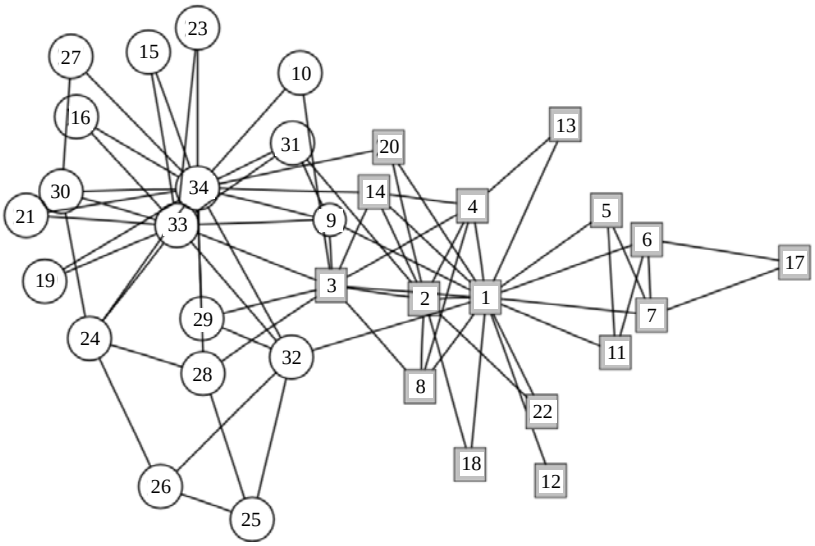

图1 Zachary空手道俱乐部成员关系网络图

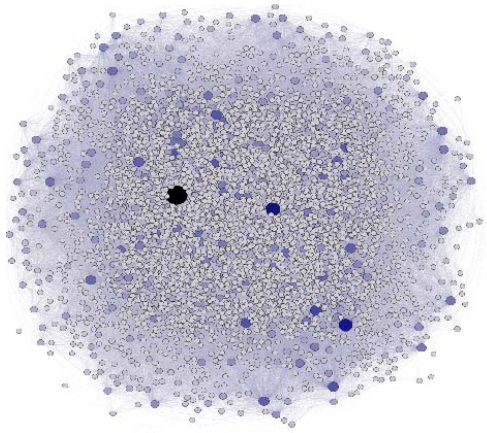

图2 邮件网络图

3.2 算法结果分析

3.2.1 算法的准确性

在实验数据集1上运行LMA算法，得到如图3所示的4个社团划分，不同颜色的节点分属不同的社团，节点越大代表该点的度值越大。与图1的真实关系网络情况进行比对，以节点1为核心的六边形和星形社团和以节点34为核心的圆和五边形社团，基本符合俱乐部分裂后的两个社团成员关系。所以LMA算法的准确性得到了验证。

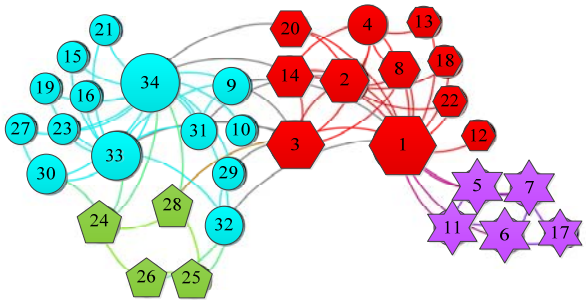

图3 空手道俱乐部社团划分结果

3.2.2 算法的有效性

在实验数据集2上运行LMA算法，发现了70个社团，与其他算法的结果比较如表1所示。

表1 社团检测算法结果

算法	发现的社团数目/个	模块度值
Newman-Girvan	65	0.486
GN	67	0.576
CNM	67	0.584
LMA	70	0.617

由实验结果可知，采用不同的算法得到的社团划分数目不同，模块度值也有区别。由于模块度值可以用来评估社团检测结果是否合理，划分是否正确。LMA算法具有最大的模块度值，所以它比其他算法划分效果更有效，结果更合理。

由于数据集2拥有4 368个节点、77 936条边，为加快算法在大型网络的计算速度，本文算法采用了灵活多任务的计算架构。在配置为Intel Core™2 Duo E4400 2 GHz，RAM 2 GB的计算机上，CNM算法需要47 min，而LMA算法用时仅4 min，所以LMA算法的计算效率更高。

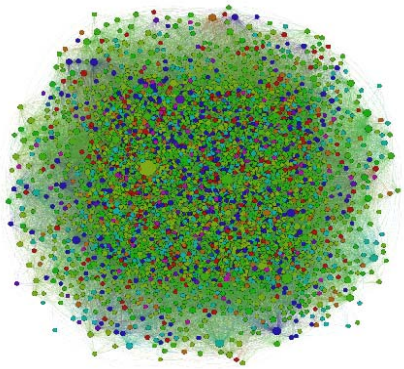

图4 邮件网络社团划分图

LMA算法的社团划分结果如图4所示，不同颜色的节点分属于不同的社团。

4 结 论

本文提出了一个动态网络社团结构演变检测模型。将随时间序列变化的动态图转变为标记有不同时间戳的静态图，使用基于相似度的LMA算法来探测静态图中的中间过程社团集合。在LMA算法结果的基础上，设置阈值并定义社团演变的关键事件，对不同时刻发现的中间过程社团进行多对多匹配。通过计算时间序列社团链的首部与中间过程社团的相似度来识别社团演变的方式。最后在真实数据集上应用该算法进行社团检测，并验证了算法的有效性。

参 考 文 献

[1] CHAKRABARTI D, KUMAR R, TOMKINS A. Evolutionary clustering[C]//Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data mining. Philadelphia: ACM, 2006.

[2] TANTIPATHANANANDH C, BERGER-WOLF T, KEMPE D. A framework for community identification in dynamic social networks[C]//Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Jose: ACM, 2007.

[3] LIN Y R, CHI Y, ZHU S, et al. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks[C]//Proceedings of the 17th International Conference on World Wide Web. Beijing: ACM, 2008.

[4] KIM M S, HAN J. A particle-and-density based evolutionary clustering method for dynamic networks[J]. Proceedings of the VLDB Endowment, 2009, 2(1): 622-633.

[5] PALLA G, BARABASI A L, VICSEK T. Quantifying social group evolution[J]. Nature, 2007, 446(7136): 664-667.

[6] ASUR S, PARTHASARATHY S. A viewpoint-based approach for interaction graph analysis[C]//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data mining. Paris: ACM, 2009, 79-88.

[7] NEWMAN M E J, GIRVAN M. Finding and evaluating community structure in networks[J]. Physical Review E, 2004, 69(2): 26-113.

[8] FENG Z, XU X, YURUK N, et al. A novel similarity-based modularity function for graph partitioning[J]. Data Warehousing and Knowledge Discovery, 2007(46-54): 385-396.

[9] KUHN H W. The Hungarian method for the assignment problem[J]. Naval Research Logistics Quarterly, 2006, 2(1-2): 83-97.

编 辑 张 俊