

一种社交网络的增量社区检测算法及实现优化

王冰玉¹, 吴振宇¹, 沈苏彬²

(1. 南京邮电大学 物联网学院, 江苏 南京 210000;
2. 南京邮电大学 计算机学院, 江苏 南京 210000)

摘要:社区检测是社交网络中常用的分析手段,目的是发现网络中联系较为紧密的节点集群,提取集群,从而进一步探索集群隐含的信息。现实中的社交网络随时间不断增大,传统的社区检测算法在不断增大的网络中运行会十分耗时,这是传统社区检测算法的一个极大弊端。针对该问题,基于传统的 K-Clique 社区检测算法,提出一种增量 K-Clique 社区检测算法。与传统 K-Clique 相比,增量 K-Clique 使用网络中新增的边和节点去更新已有的社区检测结果,而非在时间片更新时对整个网络重新进行社区检测,算法忽略极少部分的细节换取整体的高效性。实验结果表明,增量社区检测算法较传统算法在执行效率上提高显著,且检测结果与传统 K-Clique 几乎吻合。

关键词:K 团;社交网络;社区检测;增量更新

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2018)10-0064-06

doi:10.3969/j.issn.1673-629X.2018.10.013

An Incremental Community Detection Algorithm for Social Networks and Its Optimization

WANG Bing-yu¹, WU Zhen-yu¹, SHEN Su-bin²

(1. School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210000, China;
2. School of Computer Science & Technology, Nanjing University of Posts and Telecommunications,
Nanjing 210000, China)

Abstract:Community detection is a common tool to analyze social networks, aiming to discover the clusters of elements that are more closely connected in the network and then extract the clusters so as to further explore the hidden information of the cluster. As social network in real life is growing over time, using traditional community detection methods would be very time-consuming. This is a great disadvantage of the traditional community detection algorithm. In view of this, we propose an incremental K-Clique community detection algorithm based on the traditional K-Clique, which uses the edges and nodes in the new time slice to update the existing community instead of re-conducting community detection on the entire network at time slice updates. The algorithm ignores a small part of details in exchange for the overall efficiency. The experiment shows that the incremental community detection algorithm improves the execution efficiency significantly compared with the traditional one, and the detection results are almost consistent with the traditional K-clique.

Key words:K-Clique; social network; community detection; incremental updating

1 概述

在社交网络中,存在一些联系较为紧密的人群,这些人之间的互动和联系紧密和频繁,常常会把这类人群聚类用来分析^[1],在社交网络研究领域一般将这些人所构成的团体称为社区^[2]。发现社交网络中的社区是十分有意义的,可以从中挖掘出许多隐含信息^[3-4]。目前针对社交网络的社区检测算法进行了研

究,如崔泓等^[5]针对社交网络提出一种基于模块化的社区检测算法。社区的特点显而易见,即社区内部的联系紧凑,而社区之间的联系较为疏松。社区内部的人之间往往存在着一些共性,例如,在 Facebook、Twitter、Weibo 这样的社交网络中,联系紧密的人群往往在讨论一件他们都感兴趣的事件。再如在 DBLP(digital bibliography & library project, 计算机界反映共同作者

收稿日期:2017-10-19

修回日期:2018-02-26

网络出版时间:2018-05-16

基金项目:国家自然科学基金青年项目(61502246);南京邮电大学科研启动基金项目(NY215019)

作者简介:王冰玉(1993-),女,硕士研究生,研究方向为数据挖掘;吴振宇,博士,讲师,CCF 会员(61869M),研究方向为数据挖掘和人工智能;沈苏彬,博导,教授,CCF 高级会员(E200005482S),研究方向为计算网络、下一代电信网及网络安全。

网络出版地址: <http://cnki.net/kcms/detail/61.1450.TP.20180515.1655.048.html>

关系的公共数据集)这样的数据网络中,联系紧密的学者往往是研究相关或者类似的领域。在众多的社区挖掘算法中, K -clique(K 团)^[6]是一种较为经典的社区检测算法,优点在于其允许重叠社区的存在,且其参数 k 可以直观地表示社区检测的密度要求。

现实中的社交网络规模都是较大的,对 DBLP 数据集中 1990 年-2016 年中包含的会议论文的共同作者关系所形成的网络运行了 K -Clique 算法,算法的执行效率如图 1 所示。

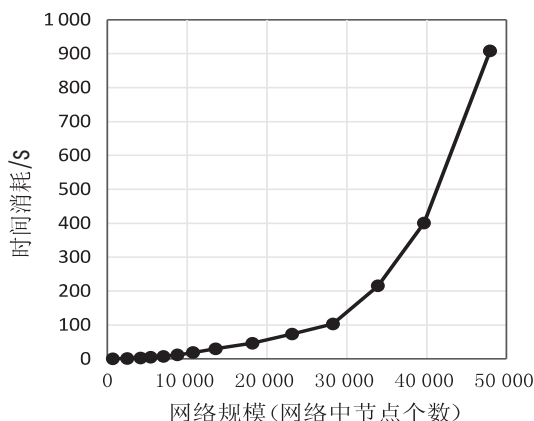


图 1 社区检测随网络规模变化的时间消耗

由图 1 可见,随着社交网络规模的不断增大, K -Clique 算法的运行时间会急剧上升。对此,文中提出一种增量式社区检测算法。该算法在时间片更新时使用新时间片上出现的节点和边去更新已有的社区检测结果,而非在整个网络上重新进行社区检测。这里的增量式是批量递增,而非每新出现一条边或一个节点时就对已有社区进行更新^[7]。也有使用滑动窗口^[8]的方式对网络进行社区检测,但是这种方式每次只能输出网络中的若干个时间切片所包含的社区,而不能针对整个网络输出其社区结构。与传统 K -Clique 进行了对比,以验证该算法的有效性。

2 相关研究

2.1 社区检测算法研究

社区检测算法有多种类型,其中比较有代表性的是基于划分的社区检测算法^[9-10]、基于模块度^[11-13]的社区检测算法、基于标签传播^[14-16]的社区检测算法以及基于团渗透^[6]的社区检测算法等。基于划分的社区检测算法基本思想是找出社区之间的链接,然后逐步删除这些链接,最后剩余的便是社区结构。2004 年 Mark Newman 基于贪心思想提出了模块度最大化的贪心算法(FN),将整体最优化问题分解为局部最优化问题。模块度的思想实际上是社区内部的边的密度大于网络中的整体密度。为了降低算法的时间复杂度, Vincent Blondel 等提出了另一种层次贪心算法^[12]—基

于标签传播(label propagation algorithm, LPA),遵从的基本思想是“在具有社区结构的网络中,任一节点都应当与其大多数邻居在同一个社区内”。重叠社区算法^[17-18]也是目前研究较多的社区检测算法,团渗透算法(clique percolation method, CPM)是重叠社区检测算法的代表性算法,这种算法将网络中共享节点的团连接在一起形成社区。这类理论认为,网络中存在的社区是可以有重叠的,这和实际的社交网络也是相符的。以微博为例,一个共同话题社区中的人可能同时也对另一个话题感兴趣。Yang 等^[19]还研究了在有向网络中利用概率模型进行社区检测的算法。

2.2 K -Clique 基本原理

K -Clique 是一种团渗透算法,基本原理可以描述如下:将图中的完全子图称为团(完全子图就是每两个点之间都有一条边的子图),若该子图的节点数为 k ,那么该子图就称为 K -Clique^[1],若两个 K -Clique 之间有 $k-1$ 个相同节点,那么就称这两个 K -Clique 是相邻的。Clique 渗透算法将彼此相邻的 clique 构成的最大集合称为社区。

3 增量式社区检测算法

3.1 算法整体设计

该算法需要对新时间片内网络中新增的元素进行处理,再使用处理结果去更新已有结果。首先定义以下几个概念:

定义 1: CSS (complete subgraph set, 完全子图集), 这个集合中存储的是已处理的网络中现有的所有完全子图结构。

定义 2: DS (difference subgraph, 差分子图), 这是一个图结构,在第一个时间片上,DS 就是当前时间片内除去 CSS 中包含元素之外的子图结构。从第二个时间片开始,DS 存储的是当前时间片中新增节点和边以及上个时间片中遗留的 DS 所共同构成的网络 US (unprocessed subgraph) 中去除了 CSS 中包含的节点及边之后的剩余元素。

定义 3: US (unprocessed subgraph, 未处理子图), 这是当前时间片中新增节点和边以及上个时间片中遗留的 DS 所共同构成的网络。

首先对第一个时间片中的网络进行传统 K -Clique 社区发现,保存第一个时间片中节点数大于等于 k 的 CSS,以及时间片中关于 CSS 的差集所构成的 DS,这个 DS 实际上也就是非完全子图所构成的局部网络,这些非完全子图虽然在当前处理的新增网络内没有被涵盖在社区之中,但有可能会与后续新增网络中的节点构成完全子图,从而参与社区更新。

在对下一个时间片的数据进行处理时,需要将 DS

也融入当前的时间片新增数据中构成当前 US。对于当前时间片上的 US,先找出其中包含的所有完全子图,之后,以新发现的完全子图去更新 CSS 以及 DS。处理完最后一个时间片中的 US 后,对于 CSS 中的每一个完全子图,如果两个完全子图之间的共同节点数大于等于 $k-1$,那么以完全子图为基础节点,在两个完全子图之间连一条边,构成一张宏观上的图 H,图 H 中的每个节点都是一个完全子图,图 H 中的每一个连通图就是整个网络中的一个社区。

上述步骤可以用 Algorithm1 来描述,其中最主要的部分包括 updateCSS、updateDS。

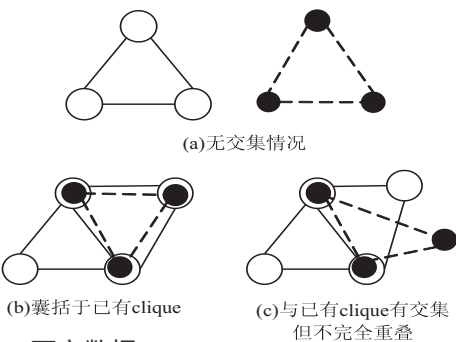
```
Algorithm1:
Input: dynamic Graph increases over time, K
Output: Communities in Graph
Get CSS in first time slice
Get DS in first time slice
While time slice:
    Get US by combining DS with new edges and nodes in current
time slice
    Find newCliques in US
    Call updateCSS( newCliques)
    Call updateDS( newCliques)
    Communities= combination of those cliques in CSS who's
common nodes>k-1 (connected parts in H)
```

3.2 更新 CSS

在处理新时间片上未处理数据 US 时分为两步,首先需要获取当前 US 中所包含的完全子图结构,然后以这些新的完全子图去更新 CSS。更新 CSS 时分为以下三种情况:

- (1) 新发现的 clique 与已有的所有 clique 均无交集;
- (2) 新发现的 clique 完全囊括于 CSS 中已有的 clique 中;
- (3) 新发现的 clique 与多个 clique 有交集,但不囊括于其中任何一个。

这几种情况如图 2 所示,白色节点和实线构成的图表示已有 clique,黑色节点和虚线构成的图表示新发现的 clique。



万方数据 2 CSS 更新类别

情况(1):直接将新发现的 clique 加入 CSS 中即可,不需要做其他改变。

情况(2):新发现的 clique 对 CSS 中已有的 clique 不会造成结构和数量上的改变,因此不需要对 CSS 进行更新。

情况(3):需要将与新 clique 有交集的已有 clique 进行结构上的更新,可能会造成 CSS 中已有 clique 的数量的变化。对于结构上可能造成的改变,采取的方式是,将新发现的 clique 与与其有交集的若干个 clique 先拼接成一张临时图结构,再从这个临时图结构找出其中包含的完全子图。以这个方式重构相关 clique 之后,需要将之前的 clique 从 CSS 中删除,然后将重构后的若干个新 Clique 加入 CSS 中。更新 CSS 的算法如下:

```
Algorithm2:updateCSS
Input:newCliques
Output:updated CSS
If CSS==null:
    CSS=newCliques;
Else:
    for clique innewCliques:
        If clique has no intersection with CSS:
            add clique into CSS;
        Else:
            If clique belongs to one of CSS:
                Do nothing;
            Else:
                Joint clique with related cliques in CSS;
                Find new cliques structure in the jointed graph;
                Add these new cliques into CSS and remove the old ones;
```

3.3 更新 DS

在更新 DS 时,基本原则是将新发现的 clique 中包含的节点和边从 DS 中删除,以免这些节点在下个时间片的 US 内被重复运算。但是实际情况下,新发现的 clique 中的节点可能还与 DS 中其他非 clique 中的节点有联系,为了保存这样的联系,需要保留这样的 clique 节点。判断的依据是,如果某个 clique 中包含的某个节点的度大于其所在的所有 clique 中所有的节点数,即说明当前节点除了存在于 clique 中的边以外还有其他边连接,这样的节点就不予删除。采用遍历的方式收集需要删除的节点,上述过程描述如下:

```
Algorithm3:updateDS
Input:newCliques
Output:updated DS
S=all nodes in newCliques
For node N in S:
    Find all the cliques innewCliques which include node N
    If degree of node in DS>=sum of all cliques size:
```

```
Remove node from S
For clique innewCliques:
Remove edges innewClique from DS
Remove nodes in S from G
```

4 性能优化

为了提高算法的执行效率,缩短节点与 clique 的查找时间,文中采用两张哈希表的方式来存储 clique 集合以及节点到已有 clique 之间的映射关系,如图 3 所示。

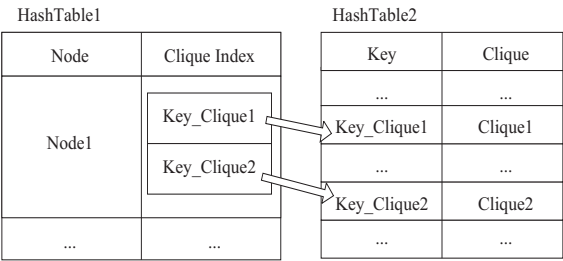


图 3 clique 存储方式

图 3 中有两张哈希表,HashTable2 中,每个 clique 都是一个真实的图结构,且每一个 clique 都有唯一对应的 key 值。HashTable1 表示节点到 clique 键值对的映射,其中的 Key_Clique 表示的就是 HashTable2 中的 key。当节点属于某一个 clique 中,那么就将 node 作为 key,该 clique 对应的编号作为值存储在左侧的哈希表中。

同时还定义了一个资源池 POOL,即 clique 的 key 值的资源分配池,定义 POOL 的目的是当容纳 clique 哈希表有更新时,能够有效地重分配 key 值。在更新 HashTable2 中的 clique 时,对 POOL 资源池的更新分为两种情况:

(1)新发现的 clique 没有对已有的 clique 的结构造成影响,只要将新发现的 clique 直接加入 Hash-Table2 中并为其分配未使用过的唯一键值作为其标识。而这个时候就必须要知道哪些键值已经被使用过,每次都重新遍历一次 HashTable2 的所有键就过于麻烦,而使用 POOL 资源池时就可以不用每次都遍历,直接从 POOL 池中选取键值即可,选取之后 POOL 资源池中对应的键值会被删除,以保证键的唯一性。

(2)另一种情况是新发现的 clique 结构对已有的 clique 结构造成了影响,这种情况下要采取的措施是,结构改变了的 clique 会从 HashTable2 中删除,并且 POOL 资源池对这些被删除的 clique 的键进行回收。随后将结构改变后的 clique 重新放入 HashTable2 中,并且从 POOL 中分配相应数目的键值对结构更新后的 clique 进行标识。这种情况下使用 POOL 的好处是,既保证了 HashTable2 中每个 clique 都有唯一标识,又

可以使得键能被有效地回收利用,不用每次去与已有 clique 比对,从而保证不会分配已经使用过的键值。

5 实验

5.1 DBLP 数据集

DBLP 是计算机界反映共同作者关系的公共数据集,数据集中包含了期刊或会议论文的共同作者信息以及论文发表时间信息。在这个数据集中,提取了 1990-2016 年发表于 ICML 会议的论文的共同作者信息,以作者为节点,以作者之间的共同发表文章的关系为边,即若两个作者共同发表了一篇文章,就在这两个作者中间连接一条边。最终的图中共包含 4 771 个节点,9 641 条边。

5.2 算法效率

分别在上述数据集中对比了传统 K-Clique 与增量 K-Clique 在 k 分别为 2,3,4,5 时的运行效率,如图 4 所示。

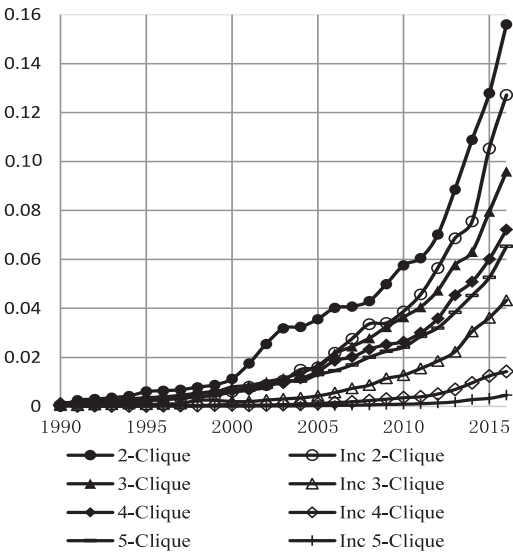


图 4 增量 K-Clique 与传统 K-Clique 效率对比

可以看到,随着时间片的递增,网络规模不断扩大,增量 K-Clique 算法比传统 K-Clique 算法在运行效率上有明显的提升,且 k 越大,效率提升得越显著。这是由于随着 k 的增大,对社区的密集程度要求更高,发现的社区数量相对较少,能够更加有效地减少社区增量更新过程中的一些计算量。

5.3 算法准确率

表 1 分别记录了 $k=2,3,4,5$ 时从 1990 年到某一年使用 K-Clique (CK) 算法和增量 K-Clique (incremental K-Clique, ICK) 算法所发现的社区数量比较。可以看到, $k=2$ 时, K-Clique 与增量 K-Clique 算法均是寻找网络中的连通图,因此两种方式寻找出的社区数量是一致的。当 $k \geq 3$ 时,由于增量式 K-Clique 算法会忽略少量的连接细节,因此发现的社区数量与 K-

Clique 相比会略多一些,但是数量基本与 K-Clique 发现的社区数量持平。

表 1 K-Clique 与增量 K-Clique 社区检测数量比较

Year	$k = 2$		$k = 3$		$k = 4$		$k = 5$	
	KC	IKC	KC	IKC	KC	IKC	KC	IKC
1990	25	25	29	29	4	4	2	2
1991	62	62	35	35	9	9	3	3
1992	77	77	41	41	10	10	3	3
1993	66	86	47	47	11	11	3	3
1994	100	100	57	57	15	15	5	5
1995	108	108	63	63	16	16	5	5
1996	120	120	72	72	17	17	5	5
1997	129	129	84	84	20	20	6	6
1998	135	135	96	96	25	25	6	6
1999	138	138	122	123	31	31	7	7
2000	167	167	140	141	42	42	10	10
2001	25	25	169	170	46	46	11	11
2002	62	62	198	199	58	58	15	15
2003	77	77	236	237	70	70	16	16
2004	66	86	278	280	90	90	18	18
2005	100	100	333	336	112	112	26	26
2006	108	108	379	382	136	136	34	34
2007	120	120	419	423	160	160	44	44
2008	129	129	469	474	182	182	55	55
2009	135	135	499	505	213	213	67	67
2010	138	138	535	541	233	233	72	72
2011	312	312	535	541	269	269	85	85
2012	335	335	612	619	320	320	103	104
2013	343	343	706	716	390	393	129	130
2014	356	356	796	808	462	465	157	158
2015	400	400	897	910	548	552	195	196
2016	415	415	1011	1025	643	649	237	239

为了更直观地看到算法的检测效果,图 5 为在检测出社区数量较小的情况下 K-Clique 算法与增量 K-Clique 算法的社区检测结果对比。取时间片 1990 至 1993 年为止的($k = 4$ 时)两种算法社区检测结果作对比,可以看到增量 K-Clique 算法与传统 K-Clique 的检测结果基本吻合。

6 结束语

社区检测是现实中各种网络的重要分析手段,因此社区检测的研究具有重要的实际意义。文中对 K-Clique 团渗透算法(CMP)在大规模网络中进行改进。这种改进算法适用于网络结构比较紧密的大规模网络

K-Clique	增量 K-Clique
‘John H. Gennari’, ‘KazuoHiraki’, ‘Yoshinobu Yamamoto’, ‘Yuichiro Anzai’	‘John H. Gennari’, ‘KazuoHiraki’, ‘Yoshinobu Yamamoto’, ‘Yuichiro Anzai’
‘DanaKedzier’, ‘DonaldMichie’, ‘ClaudeSammut’, ‘Scott Hurst’	‘DanaKedzier’, ‘DonaldMichie’, ‘ClaudeSammut’, ‘Scott Hurst’
‘Paul R. Cohen’, ‘Adam St. Amant’, ‘Adam Carlson’, ‘Lisa Ballesteros’	‘Paul R. Cohen’, ‘Adam St. Amant’, ‘Adam Carlson’, ‘Lisa Ballesteros’
‘James Garrett’, ‘Bradley L. Whitehall’, ‘Thomas G. Dietterich’, ‘Stephen C. Y. Lu’, ‘Richard J. Doyle’, ‘BrianFalkenhainer’, ‘Steve A. Chien’	‘James Garrett’, ‘Bradley L. Whitehall’, ‘Thomas G. Dietterich’, ‘Stephen C. Y. Lu’, ‘Richard J. Doyle’, ‘BrianFalkenhainer’, ‘Steve A. Chien’
‘JohnVittal’, ‘Bernard Silver’, ‘William J. Frawley’, ‘Kelly Bradford’, ‘Glenn A. Iba’	‘JohnVittal’, ‘Bernard Silver’, ‘William J. Frawley’, ‘Kelly Bradford’, ‘Glenn A. Iba’
‘David K. Tcheng’, ‘Stephen C. Y. Lu’, ‘Larry A. Rendell’, ‘Bruce L. Lambert’	‘David K. Tcheng’, ‘Stephen C. Y. Lu’, ‘Larry A. Rendell’, ‘Bruce L. Lambert’
‘Lee A. Appelbaum’, ‘Stuart L. Crawford’, ‘Richard M. Tong’, ‘Robert M. Fung’	‘Lee A. Appelbaum’, ‘Stuart L. Crawford’, ‘Richard M. Tong’, ‘Robert M. Fung’
‘Stewart W. Wilson’, ‘Alexandre Parodi’, ‘PierreBonelli’, ‘Sandip Sen’	‘Stewart W. Wilson’, ‘Alexandre Parodi’, ‘PierreBonelli’, ‘Sandip Sen’
‘Glenn R. Koller’, ‘Qian Yang’, ‘Jerry B. Weinberg’, ‘Gautam Biswas’	‘Jerry B. Weinberg’, ‘Qian Yang’, ‘Glenn R. Koller’, ‘Gautam Biswas’
‘Davide De Marchi’, ‘Attilio Giordana’, ‘Filippo Brancadori’, ‘FrancescoBergadano’, ‘Lorenza Saitta’	‘Davide De Marchi’, ‘Attilio Giordana’, ‘Filippo Brancadori’, ‘FrancescoBergadano’, ‘Lorenza Saitta’
‘Christopher M. Tuck’, ‘John E. Laird’, ‘Eric S. Yager’, ‘MichaelHucka’	‘Christopher M. Tuck’, ‘John E. Laird’, ‘Eric S. Yager’, ‘MichaelHucka’

图 5 K-Clique 与增量 K-Clique 社区检测结果比较

中,在算法效率上有明显提升。而在网络规模稀疏的情况下可能效果会有所衰减,因为算法可能会忽略较多的细节从而导致准确率下降。后续会将这种算法应用到实际问题中,例如挖掘社交网络的密集用户社区,以进一步挖掘社区中的隐含信息等。

参考文献:

[1] NEWMAN M E, PARK J. Why social networks are different from other types of networks[J]. Phys Rev E Stat Nonlin Soft Matter Phys, 2003, 68(3 Pt 2):036122.

[2] 王 莉,程学旗. 在线社会网络的动态社区发现及演化[J]. 计算机学报,2015,38(2):219-237.

[3] 范超然,黄曙光,李永成. 微博社交网络社区发现方法研究[J]. 微型机与应用,2012,31(23):67-70.

[4] 许 为,林柏钢,林思娟,等. 一种基于用户交互行为和相似度的社交网络社区发现方法研究[J]. 信息安全,2015(7):77-83.

[5] 崔 泓. 社交网络中一种基于模块化的社区检测算法[J]. 计算机工程,2014,40(7):62-68.

[6] PAIIA G, DERENYI I, FARKAS I, et al. Uncovering the overlapping community structures of complex networks in nature and society[J]. Nature,2005,435(7043):814-818.

[7] DUAN Dongsheng, LI Yuhua, LI Ruixuan, et al. Incremental K-clique clustering in dynamic social networks[J]. Artificial Intelligence Review,2012,38(2):129-147.

[8] CHAKRABARTI D, KUMAR R, TOMKINS A. Evolutionary clustering[C]//Twelfth ACM SIGKDD international conference on knowledge discovery and data mining. Philadelphia, PA, USA:ACM,2006:554-560.

[9] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. Proc of National Academy of Science,2002,9(12):7821-7826.

[10] TYLER J R, WILKINSON D M, HUBERMAN B A. Email

as spectroscopy:automated discovery of community structure within organizations[M]//Communities and technologies. Netherlands:Springer,2003:143-153.

[11] NEWMAN M E. Fast algorithm for detecting community structure in networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics,2003,69(6Pt2):066133.

[12] NEWMAN M E J, GIRVAN M. Finding and evaluating community structure in networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004, 69 (2Pt2): 026113.

[13] LU Hao, HALAPPANAVAR M, KALYANARAMAN A. Parallel heuristics for scalable community detection[J]. Parallel Computing,2015,47:19-37.

[14] 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(3Pt2):036106.

[15] LEUNG I X, HUI P, LIÒ P, et al. Towards real-time community detection in large networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2009, 79 (2): 066107.

[16] TIBÉLY G, KERTÉSZ J. On the equivalence of the label propagation method of community detection and a Potts model approach[J]. Physica A Statistical Mechanics & Its Applications,2008,387(19-20):4982-4984.

[17] 乔少杰,韩 楠,张凯峰,等. 复杂网络大数据中重叠社区检测算法[J]. 软件学报,2017,28(3):631-647.

[18] 李刘强,桂小林,安 健,等. 采用模糊层次聚类 的社会网络重叠社区检测算法[J]. 西安交通大学学报,2015,49(2):6-13.

[19] YANG T, CHI Y, ZHU S, et al. Directed network community detection:a popularity and productivity link model[C]//SIAM international conference on data mining. Columbus, Ohio, USA:[s. n.],2010:742-753.

(上接第 52 页)

(8):1091-1094.

[7] BINDEL S, HILT B, CHAUMETTE S. F-ETX:an enhancement of ETX metric for wireless mobile networks[M]//Communication technologies for vehicles. [s. l.]:[s. n.], 2015:35-46.

[8] JAGADEESH G R, SRIKANTHAN T. Online map-matching of noisy and sparse location data with hidden Markov and route choice models[J]. IEEE Transactions on Intelligent Transportation Systems,2017,18(9):2423-2434.

[9] 崔青华,夏井新. 基于 ARIMA-GARCH 模型的城市主干道行程时间时变置信区间预测[J]. 东南大学学报:英文版,2014,30(3):358-362.

[10] 王 兴,蔡 毅. 改进的无线 Mesh 网路由判据算法及其路由协议[J]. 计算机系统应用,2016,25(5):153-158.

[11] 宋路力,王力数据. Markov 模型的移动对象位置预测算法研究

[D]. 徐州:中国矿业大学,2016.

[12] 杨 迪. 基于混合多步 Markov 模型的位置预测方法研究[D]. 沈阳:东北大学,2014.

[13] 王黎明,玉 连,杨 楠. 应用时间序列分析[M]. 上海:复旦大学出版社,2008.

[14] 郭自强. 基于地理位置预测的 DSR-iETT 路由协议研究[D]. 沈阳:东北大学,2013.

[15] 杨奔全. 基于位置预测的 Ad Hoc 网络路由协议研究[D]. 长沙:湖南大学,2010.

[16] 丁 雄,卢 嫣. 基于 MESH 网络环境下的负载均衡路由判据研究[J]. 信息系统工程,2016(5):129.

[17] 符 琦. 一种具有业务感知的多路径 QoS 路由策略[J]. 计算机学报,2014,37(10):2153-2164.

[18] 陈贵海,李宏兴,韩 松,等. 多跳无线网络中基于网络编码的多路径路由[J]. 软件学报,2010,21(8):1908-1919.