第35卷 第12期 计 算 机 工 程 2009年6月
Vol.35 No.12 Computer Engineering June 2009
·博士论文· 文章编号：1000—3428(2009)12—0029—03 文献标识码：A 中图分类号：TP311.13

可多边并行移出的社团发现方法

熊中敏[1,2]，黄冬梅[2]
(1. 复旦大学计算机与信息技术系，上海 200433；2. 上海海洋大学信息学院，上海 201306)

摘 要：针对GN算法计算效率低下的缺陷，提出一个基于边的中介值测度的发现网络潜在社团结构的新算法。该算法在完成所有边的中介值计算后，利用成分的独立性，采用并行移出各个成分中具有最大中介值的边的方法。通过理论分析，在作为实验测试平台的实际的数据集上进行实验验证，结果表明该算法是快速、有效的。

关键词：社团发现；社会网络；社团结构；图挖掘

Community Detection Method with Multi-edge Simultaneous Removal

XIONG Zhong-min[1,2], HUANG Dong-mei[2]
(1. Department of Computing and Information Technology, Fudan University, Shanghai 200433;
2. School of Information, Shanghai Ocean University, Shanghai 201306)

【Abstract】To address the slow speed of GN algorithm, a new algorithm based on betweenness scores of edges is presented for detecting the underlying community structure in networks. Employing component independency, this algorithm presents a new method through which all edges with the highest betweenness score in respect of each component is simultaneously removed when all betweenness scores are computed. It is proved that this algorithm is fast and effective through theoretical analysis and experiments with several real data sets which are acted as test beds.

【Key words】community detection; social networks; community structure; graph mining

1 概述

社会网络中社团发现的层次聚类方法分为2类：聚合和分裂。由于聚合的方法发现不了社团的外围成员，因此提出了分裂的方法。最有名的一个算法是文献[1]提出的GN算法，但一条边移出后，剩余的边要全部重计算中介值，此算法效率低下。为此提出了一些改进算法，比如文献[2]提出的Monte Carlo 重采样方法，文献[3]提出的基于统计最小环的方法，文献[4]提出的基于 optimal modularity 的方法。由文献[5]可知，尽管这些方法提高了效率，但分析质量却低于GN算法。

通过文献[1]可知，网络发生分离时，从一个成分中移出边并不影响另一成分中边的中介值，具有显著社团结构的网络在GN算法处理初期就会发生成分分离，利用此现象提高GN 算法的效率是个悬而未决的问题。据笔者所知，目前未见任何关于利用此现象进行挖掘算法深入研究的报道。

2 中介值和GN算法

GN 算法处理的是具备简单顶点和无向、无权重简单边的网络，本文亦如此。GN 算法的基本思想是发现并移出网络中具有最大中介值的边。提出边的中介值的启发原理是：社团内的联结边远多于社团之间的联系边，如果要完成从一个社团到另一个社团的一次信息传递，至少要通过一条中间的联系边。逐步移出具有最大中介值的边，网络就逐步划分出单个的社团。

边的中介值的计算基于顶点之间的最短路径的统计：找出所有顶点对之间的最短路径，然后统计每条边被纳入的次数。文献[1]在图的宽度优先搜索算法的基础上提出了一个新的高效计算方法：首先计算图中一个顶点出发的所有最短路径对所有边的中介值的贡献，然后统计所有的顶点得到所有边的总的中介值。

GN 算法发现社团结构的步骤如下：

(1)从给定网络的一个顶点出发计算对边的中介值的贡献量，然后统计所有顶点计算所有边的总的中介值。

(2)发现具有最大中介值的边并移出。

(3)重计算剩余边的中介值。

(4)重复步骤(2)，直到所有边移出。

3 算法的理论基础

例1 比较GN算法和新方法对图1所示网络的分析。

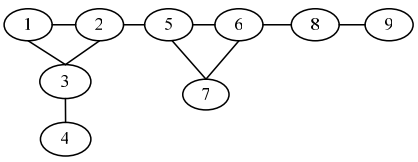

图1 一个简单网络

如前所述处理步骤，GN 算法分析结果如下：

(1)移出边(2,5)；

基金项目：国家科技部海洋公益性行业科研专项经费基金资助项目(200805016)；国家“973”计划基金资助项目(2005CB321905)；国家自然科学基金资助项目(60303008)；上海海洋大学博士启动基金资助项目

作者简介：熊中敏(1972－)，男，讲师、博士后，主研方向：数据库，数据挖掘；黄冬梅，教授

收稿日期：2008-12-04 **E-mail：**zhmxiong@fudan.edu.cn

(2)移出边(6,8)；
(3)移出边(3,4)；
(4)移出边(5,6)；
(5)移出边(5,7)；
(6)移出边(1,2)；
(7)移出边(1,3)；
(8)移出边(8,9)；
(9)移出边(6,7)；
(10)移出边(2,3)。

图 2 为 GN 算法生成的层次树结构(dendrogram)。

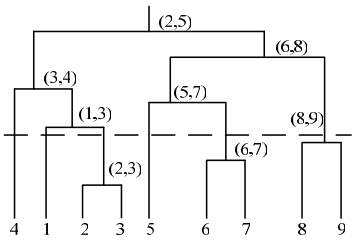

图 2　GN 算法分析例 1 生成的层次树

图 1 中的初始网络是一个连通图，将其看作新方法分析时的第一个成分。当具有最大中介值的边(2,5)移出时，图分裂为 2 个成分，它们将各自独立计算所属边的中介值并移出各自的最大中介边。最后需要将移出的边的移出次序重新调整到与 GN 算法一致，以便保持 GN 算法的分析质量高的特点。

利用上述方法分析例 1，移出图 1 中边的次序为:

(1)移出边(2,5)；
(2)移出边(3,4)和(6,8)；
(3)移出边(1,2), (5,6)和(8,9)；
(4)移出边(1,3)和(5,7)；
(5)移出边(2,3)和(6,7)。

最后需要从第一分裂层次开始比较同一层次中移出边的中介值，将具有较小中介值的边移到下一层次中。

定义 1(移出边的直接孩子) *e* 为从图 *G* 中成分 *R* 移出的一条边，若边 *f* 满足以下条件，则称之为 *e* 的直接孩子:

(1)当 *e* 移出时未导致 *R* 分裂，*f* 为 *R* 中移出的下一条边。

(2)当 *e* 移出时导致 *R* 分裂为 2 个更小的成分，*f* 为其中任一成分中首次移出的边。

直接孩子关系是可传递关系，可扩展为如下定义。

定义 2(移出边的孩子) *e* 为从图 *G* 中成分 *R* 移出的一条边，若边 *f* 为 *e* 的直接孩子或 *f* 为 *h* 的直接孩子而 *h* 为 *e* 的直接孩子，则称 *f* 为 *e* 的孩子。

一旦一条边移到下一个层次中，它的所有孩子都要下移一个层次。最后调整的移出层次次序为:

(1)移出边(2,5)；
(2)移出边(6,8)；
(3)移出边(3,4)；
(4)移出边(1,2), (5,6)；
(5)移出边(5,7)；
(6)移出边(1,3), (6,7)和(8,9)；
(7)移出边(2,3)。

按本文的处理方法分析例 1，得到如图 3 所示的层次树。在图 3 的第 5 层中，边(1,3), (6,7)和(8,9)具有相同的中介值，所以保留在同一个层次中，社会网络传统的分析方法通常这样处理。传统的社会网络分析方法常用水平线来切割层次树，交叉处表示该水平高度处网络的社团层次结构。所以移出边在层次树中的高度影响社团挖掘的分析质量。图 2 中 GN 算法所得层次树无法取得同样精确的结果。

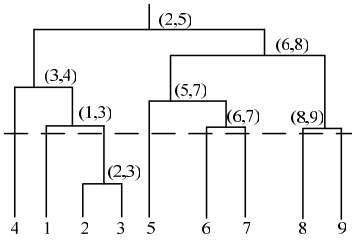

图 3　本文方法分析例 1 生成的层次树

标记 1(betweenness-bfs) 从第 2 节所述可知，计算边的中介值的基础是以某一顶点为源点，进行图的宽度优先扫描，得到源点到其他点的最短路径，并统计此扫描结果对边的总的中介值的贡献量。下文将以点 *A* 为源点的这种扫描计算记作：betweenness-bfs (*A*)。

定理 1 R_A 和 R_B 表示任意 2 个从图 *G* 中分裂出的成分，R_A 和 R_B 在计算所属边的中介值时是相互独立的，并且它们移出边、分裂为更小成分的过程也是相互独立的。

证明：通过图论中成分的定义可知，若 *R* 为图 *G* 中的一个成分，则 *R* 为 *G* 的一个极大连通子图。也就是 *R* 中任意 2 个顶点必可通过路径可达，如果顶点 *A* 可为 *R* 中任一顶点到达，则 *A* 必属于成分 *R*；否则，*A* 必不属于 *R*。

因此，一旦 R_A 中某一顶点 *A* 作为源点并执行 betweenness-bfs (*A*)，则此搜索中可到达的所有顶点必属于 R_A；反之，则不属于 R_A。这样，只有属于 R_A 的边的中介值才会受到 betweenness-bfs (*A*)的影响，而 R_B 中的边则不受影响。因此一旦 *G* 中所有顶点都执行了这样的动作，所有的边都重计算了总的中介值，R_A 和 R_B 可以分别独立地找到自己的最大中介值边并移出。证毕。

定理 2 *R* 表示任意一个从图 *G* 中分裂出的成分，*A* 表示 *R* 中任意一个顶点。如果执行 betweenness-bfs (*A*)并且顶点 *B* 可在此搜索中到达，则 *B* 必属于 *R*；否则，*B* 必不属于 *R*。

证明：此结论已在定理 1 的证明过程证明。证毕。

若 *R* 移出一条边，2 个顶点表示为 *from* 和 *to*。如果执行 betweenness-bfs (*from*)而且 *to* 不能在此搜索中到达，则此边移出必导致 *R* 分裂。

定理 3　*R* 表示任意一个从图 *G* 中分裂出的成分，*A* 表示 *R* 中任意一个顶点。如果执行 betweenness-bfs (*A*)并且边 *e* 可在此搜索中到达，则 *e* 必属于 *R*；否则，*e* 必不属于 *R*。

证明：以 *B, C* 分别表示 *e* 的 2 个顶点。(1)*e* 可在 betweenness-bfs (*A*)中到达，由定理 2 可知，*B, C* 必属于 *R*，即 *e* 属于 *R*。(2)*e* 不能在 betweenness-bfs (*A*)中到达，则 *e* 的顶点 *B, C* 必不能在 betweenness-bfs (*A*)中到达。由定理 2 可知，*B, C* 必不属于 *R*，即 *e* 不属于 *R*。由(1)和(2)可知结论成立。证毕。

4　算法描述及分析

记图为 *G*={*V, E*}，*V* 为顶点集，*E* 为边集，算法步骤如下所示:

Step1　初始化

原始图是无向、无权重的连通图并作为 Q 的第一个初始成分；
初始化分裂层次 split_Step=0；

for each a∈V(G) do passive(a)=0;

Step2 各个成分独立进行社团分裂

while Q≠∅ do {

//Step2.1: 计算当前图 G 中边的中介值

for each source∈V(G) do {若 passive(source)≠1，执行 betweenness-bfs(source)；否则，跳过此点并置 passive(source)=0}

//Step2.2: 各成分独立发现最大中介值的边并移出

for 每个初始成分 R∈Q 且 R 非当前新增成分 do {

找到 R 中最大中介值边 maxedge，首、末顶点标识为 from, to，并从图 G 中移出；假定此移出导致 R 分裂，置 flag=1；

R 中每条边的中介值初始化为 0；

maxedge 插入到 split_Step 中，若 R.former_removal_edge≠NULL，则标识 maxedge 为其孩子并置 R.former_removal_edge=maxedge；

passive(from)=1，执行 betweenness-bfs(from)并做处理：记录扫描到的边集为 edges_from；若扫描到点 to，则 R 未分裂，置 flag=0；

if flag==0 then {maxedge 从 R 中移出}

else {R 已分裂将其移出 Q；passive(to)=1，执行 betweenness-bfs(to)并做处理：记录扫描到的边集为 edges_to；

若 edges_from 非空，则作为 R 的子成分 R_a 插入到 Q 并置 R_a.former_removal_edge=maxedge；否则，点 from 从 R 中移出；

若 edges_to 非空，则作为 R 的子成分 R_b 插入到 Q 并置 R_b.former_removal_edge=maxedge；否则，点 to 从 R 中移出；

}}split_Step+=1; }

Step3 重新调整 Step2 中移出边的层次位置

for 每条边 e∈ E(G) do offset(e)=0;

for (i=0; i<H; i++) do {

找到表 $List_i$ 中 offset=0 的所有边中最大中介值 Max；

for 每条边 e∈$List_i$ do {

若 offset(e) ==0 且 e 的中介值小于 Max，则 e 移出 $List_i$ 且插入到 $List_{i+1}$，记 e.c1, e.c2 为其孩子置 offset(e.c1)+=1、offset(e.c2)+=1；

若 offset(e)≠0，则 e 移出 $List_i$ 且插入到 $List_{i+\,offset(e)}$，记 e.c1, e.c2 为其孩子置 offset(e.c1)+= offset(e)、offset(e.c2)+= offset(e)、offset(e)=0；}}

在 Step1 中，*Q* 表示目前从网络中分裂出的成分集；为了提高 Step2 的处理效率，定义变量 *passive*， *passive*(*A*)=1 表示顶点 *A* 不执行 betweenness-bfs(*A*)。

为了便于 Step3 中移出边层次的调整，在 Step2 中初始化一个列表 $List_{split\_Step}$，用来包含当前 *split_Step* 中移出的边；为每个成分定义变量 *former_removal_edge*，用来记录上次移出的边。

Step2 根据定理 2 判定成分分裂；根据定理 3 将分裂出的子成分更新集合 *Q*。在 Step3 中，*offset*(*e*)表示边 *e* 下移的位移量；*H* 表示 Step2 中 *split_Step* 最大值。

设图 *G* 有 *m* 条边、*n* 个顶点，Step1 耗时 *O*(*n*)；Step3 耗时 *O*(*H*×*n*)，*H* 小于 *m*，故耗时为 *O*(*mn*)；由文献[1]可知 *n* 个顶点执行 betweenness-bfs 需耗时 *O*(*m*×*n*)，设 Step2 循环 *iterations* 次，算法时间复杂度决定于 *O*(*iterations*×*m*×*n*)。最坏情况下，Step2 中每个成分每次移出一条边，且一旦分裂只分离出一个顶点，即 *Q* 总保持一个成分，*iterations* 等于 *G* 的边数 *m*，算法总耗时 $O(m^2n)$，等同于 GN 算法。最好情况下，*Q* 中每个成分移出一条边时导致分裂为 2 个更小的子成分，最后所得社团层次结构树为完全二叉树，设其高度为 *K*，有 *K*=lb*n*-1，故总耗时为 *O*(*mn*lb*n*)。一般情况下，新算法效率高于 GN 算法。

5 实验结果

利用现有文献[1-5]中用作试验台的实际数据集，比较了新算法和 GN 算法的执行性能，见表 1。算法用 C++实现，运行在 1.86 GHz Intel Celeron(R)处理器、512 MB 内存的 DELL 笔记本电脑上。

表 1 新方法和 GN 算法执行效率比较

网络	顶点数 *n*	边数 *m*	运行时间/s		提高比率/(%)
			GN 算法	新算法	
karate	34	78	0.040 5	0.030 2	25.43
dolphins	62	159	0.163 9	0.116 0	29.23
football	115	616	4.076 3	3.171 6	22.19
lesmis	77	254	0.528 0	0.421 4	20.19
netscience	1 589	2 742	31.124 2	5.517 0	82.27
polbooks	105	441	1.944 5	1.361 3	29.99

参考文献

[1] Newman M E J, Girvan M. Finding and Evaluating Community Structure in Networks[J]. Phys. Rev. E., 2004, 69(2): 113-127.

[2] Tyler J R, Wilkinson D M, Huberman B A. Email as Spectroscopy: Automated Discovery of Community Structure Within Organizations[C]//Proc. of the 1st Int'l Conf. on Communities and Technologies. Amsterdam, Holland: [s. n.], 2003: 81-96.

[3] Radicchi F, Castellano C, Cecconi F, et al. Defining and Identifying Communities in Networks[C]//Proc. of the National Academy of Science of USA. Rome, Italy: [s. n.], 2004: 2658-2663.

[4] Newman M E J. Fast Algorithm for Detecting Community Structure in Networks[J]. Phys. Rev. E., 2004, 69(6): 133-137.

[5] Newman M E J. Detecting Community Structure in Network[J]. The European Physical Journal, 2004, 38(2): 321-330.

编辑 任吉慧

(上接第 28 页)

结果可以看出，该方法能自动地挖掘出网络中的显著流量，无需先验知识，无需预先设置过滤器。P2P 疑似度的指标可以准确地判别了显著流量的性质。

5 结束语

本文提出一种基于多维聚类挖掘的频繁项挖掘方法。相比于传统的流分析方法，主要有 3 个特点：(1)自动地挖掘网络中的显著流量，无需事先定义过滤器；(2)对显著流量进行了多维描述；(3)对 IP 维进行了层次聚类，能够反映 IP 子网的流量情况。在多维聚类的基础上，提出显著流的 P2P 疑似度指标，并利用这些判定规则实现了未知 P2P 流量的识别。

参考文献

[1] 陈海军, 李仁发, 杨 磊. 基于 Linux 内核扩展模块的 P2P 流量控制[J]. 计算机工程, 2007, 33(1): 87-89.

[2] Estan C. Automatically Inferring Patterns of Resource Consumption in Network Traffic[C]//Proc. of the 2003 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications Table of Contents. Berlin, Germany: [s. n.], 2003.

[3] Miller D. Polymorphic Worm Detection and Defense: System Design, Experimental Methodology, and Data Resources[C]//Proc. of the IEEE Int'l Conf. on Large-scale Attack Defense. Pisa, Italy: [s. n.], 2006.

编辑 陈 文