基于局部探测的快速复杂网络聚类算法

金 弟,刘大有,杨 博,刘 杰,何东晓,田 野
(1.吉林大学计算机科学与技术学院,吉林长春 130012; 2.吉林大学符号计算与知识工程教育部重点实验室,吉林长春 130012)

摘 要: 目前复杂网络的规模越来越庞大,且呈现天然的分布式特性,因此从局部观点出发提出快速网络聚类算法就成为迫切需要.为解决这一问题,本文基于对网络模块性函数 Q 的分析,推导出一个针对于单个结点的局部目标函数 f,并证明 Q 函数随网络中任一结点的 f 函数呈单调递增趋势,进而提出一个基于局部优化的近线性网络聚类算法 FNCA.在该算法中,每个结点仅利用网络的局部簇结构信息来优化自身的目标函数 f,所有结点通过相互协同来实现对整个网络的聚类.通过计算机生成网络和真实网络对算法 FNCA 进行测试,实验表明,该算法的运行效率和聚类质量都要明显优于当前的一些优秀网络聚类算法.

关键词: 复杂网络; 网络聚类; 簇结构; 局部探测
中图分类号: TP18; TP391 **文献标识码:** A **文章编号:** 0372-2112 (2011) 11-2540-07

Fast Complex Network Clustering Algorithm Using Local Detection

JIN Di, LIU Da-you, YANG Bo, LIU Jie, HE Dong-xiao, TIAN Ye
(1. *College of Computer Science and Technology, Jilin University, Changchun, Jilin* 130012, *China*;
2. *Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin* 130012, *China*)

Abstract: Recently, complex networks are always very huge and take on distributed nature. Therefore it is gradually becoming instant requirement to propose fast network clustering algorithms in the sight of local view. For the problem, this paper deduces a local objective function f aiming to each node in the network, which is based on the profound analysis on network modularity function Q, and proves that Q is monotone increasing with function f of any node, and then proposes a fast network clustering algorithm (FNCA) by using local optimization. In this algorithm, each node optimizes its own objective function f by only local information, and all the nodes collectively optimize function Q to detect network community structure. Both efficiency and effectiveness of algorithm FNCA are tested against computer-generated and real-world networks. Experimental result shows that this algorithm is better than some excellent network clustering algorithms in term of these two respects.

Key words: complex network; network clustering; community structure; local detection

1 引言

现实世界中的许多复杂系统都以复杂网络的形式存在,或者能被转化成复杂网络进行处理,如社会网、生物网、Web 网络、科技网络等.网络簇结构是复杂网络最普遍和最重要的拓扑结构属性之一,具有同簇结点相互连接紧密、异簇结点相互连接稀疏之特点[1].随着应用领域的不同,簇结构具有不同的内涵,譬如:社会网中的社区代表了具有某些相近特征的人群,生物网络中的功能组揭示了具有相似功能的生物组织模块,Web 网络中的文档类簇包含了大量具有相关主题的 Web 文档,等等.复杂网络聚类(又称社区挖掘)的目的就是要探测并揭示出异构复杂网络中固有的簇结构.虽然网络聚类与普通数据聚类[2]有着相近之处,但它们却存在着本质区别.

复杂网络聚类研究具有十分重要的理论及现实意义,他不仅吸引了大量不同学科的研究工作者,而且已被应用于如恐怖组织识别、蛋白质功能预测、基因调控网络构建、Web 社区挖掘、链接预测等众多领域中[3,4].

目前已有许多各具特色的网络聚类算法被提出,按照文献[3]的观点,他们中的大多数可按照所采用的基本求解策略,归纳为两大类:启发式方法和基于优化的方法.其中,基于启发式的方法有:算法 GN(Girvan-Newman)[1]、算法 CPM(Clique Percolation Method)[5]、算法 FEC(Finding and Extracting Communities)[6]、算法 LPA(Label Propagation Algorithm)[7]、算法 ACOMRW(Ant Colony Optimization with Markov Random Walk)[8]等等.在基于优化的

收稿日期:2010-12-14;修回日期:2011-04-21
基金项目:国家自然科学基金(No.60773099,60703022,60873149,60973088);国家 863 项目(No.2006AA10Z245);模式识别国家重点实验室开放课题(No.09-1-1);中央高校基本科研业务费专项资金(No.200903177);复旦大学智能信息处理上海市重点实验室开放课题(No.IIPL-09-007)

方法中,将网络模块性函数 Q[9]作为目标函数目前已被多数研究者所广泛接受,这一类型的方法有:算法 FN(Fast Newman)[10]、算法 SA(Simulated Annealing)[11]、算法 LPAm(Modularity-Specialized Label Propagation Algorithm)[12,13]、算法 GALS(Genetic Algorithm with Local Search)[14]等等.此外,文中算法 FNCA 也属于这类方法.

一方面,当前基于优化 Q 函数的网络聚类算法大都是从全局观点出发,以全局网络拓扑结构作为研究对象来实现对 Q 函数的优化,其时间效率一般都不够理想;另一方面,随着计算机科学技术和交叉研究的迅速发展,众多规模越来越大、具有分布式特性的各种复杂网络呈现在研究者的面前[3].因此,从局部观点出发提出快速、并行的网络聚类算法就成为人们亟待解决的问题.基于此,本文通过分析 Q 函数(面向整个网络的目标函数),推导出一个具有局部特征的 f 函数(针对每个结点的目标函数),并证明 Q 函数随网络中任一结点的 f 函数单调递增,进而提出一个基于 f 函数的快速(近线性)网络聚类算法 FNCA(Fast Network Clustering Algorithm).在算法 FNCA 中,网络的每个结点仅利用其局部簇结构信息来优化自身的 f 函数,并通过所有结点的协同作用来实现对 Q 函数的优化.实验结果及与相关算法的对比分析表明了 FNCA 的有效性.

2　算法 FNCA

2.1　问题定义

2004 年,Newman 等[9]基于对"网络簇结构越明显,它与随机网络的差异就越大"这一直观现象的思考,提出了一个可定量评价网络簇结构优劣的度量标准,被称为网络模块性函数(Q),其目前已被大多数相关领域的学者所接受.Q 函数的定义为:"网络簇内实际的边数"与"完全随机的连接情况下簇内期望的边数"之差.

给定一个无向无权网络 $N(V,E)$,假设点集 V 被划分(聚类)为若干个类簇.若网络中任一结点 i 的标签为 $r(i)$,他所属的类簇为 $c_{r(i)}$,则 Q 函数可被定义为

$$Q=\frac{1}{2m}\sum_{ij}\left(\left(A_{ij}-\frac{k_ik_j}{2m}\right)\times\delta(r(i),r(j))\right) \quad (1)$$

其中 $A=(A_{ij})_{n\times n}$ 表示网络 N 的邻接矩阵,如果结点 i 与结点 j 之间有边连接,则 $A_{ij}=1$,否则 $A_{ij}=0$;对于函数 $\delta(u,v)$,如果 $u=v$,他取值为 1,否则取值为 0;k_i 表示结点 i 的度,被定义为 $k_i=\sum_j A_{ij}$;$m=\frac{1}{2}\sum_{ij}A_{ij}$,表示网络 N 中总的边数.

将式(1)改写成式(2),其中所有符号的含义都与式(1)相同.

$$Q=\frac{1}{2m}\left\{\sum_{ij}(A_{ij}\times\delta(r(i),r(j)))-\sum_{ij}\left(\frac{k_ik_j}{2m}\times\delta(r(i),r(j))\right)\right\} \quad (2)$$

可以看出,式(2)中花括号内减号的左边表示网络簇内实际的连接数目,减号右边表示随机连接情况下簇内的期望连接数目,很显然他们的差(即 Q 函数)就可以度量一个网络聚类结果所对应簇结构的优劣.文中算法的目标就是对复杂网络进行聚类,使得聚类结果对应的 Q 函数值最大.

2.2　算法主要思想

目前基于 Q 函数的网络聚类算法大都是从全局角度进行分析,通过设计一种有效的搜索策略来实现对 Q 函数的优化,由于这些算法进行每步搜索时都需要用到整个网络的拓扑结构信息,所以他们的运算效率一般都不是很高.针对这一问题,不同于已有方法,本文试图从局部观点出发,使网络中每个结点独立计算并优化自身的局部函数,并通过所有结点协同作用来实现对 Q 函数的优化.为实现上述思想,下面我们首先对 Q 函数进行分析.

我们将式(1)转化为式(3),即把 Q 函数表示为网络中所有结点的 f 函数之和.很显然,f 函数可理解为:从网络中任一结点的局部观点来看,"簇内实际连接数目"与"随机连接情况下簇内期望连接数目"之差,所以网络中每个结点的 f 函数都可以从一个局部角度来度量网络簇结构的优劣.下面给出 f 函数的相关性质和定理.

$$Q=\frac{1}{2m}\sum_i f_i,\ f_i=\sum_{j\in C(i)}\left(A_{ij}-\frac{k_ik_j}{2m}\right) \quad (3)$$

性质 1　对于 $\forall i\in V$,复杂网络中结点 i 的局部函数 f_i 仅与他所在的簇 $c_{r(i)}$ 相关.

定理 1　对于 $\forall i\in V$,复杂网络中的全局函数 Q 随结点 i 的局部函数 f_i 呈单调递增趋势.

证明　不妨设网络 $N(V,E)$ 的当前簇结构为 C,取 $\forall i\in V$,假设结点 i 的标签由 $r(i)$ 变为 $r(j)$,使得当前网络簇结构变为 C'.若 $r(i)\neq r(j)$,那么在 C' 中,结点 i 原来所在的簇将变为 $c'_{r(i)}=c_{r(i)}-\{i\}$,结点 i 当前所在的簇将变为 $c'_{r(j)}=c_{r(j)}\cup\{i\}$.

由式(3)可知,对于任一结点,如果他所在的簇发生了变化,那么他所对应的 f 函数值也会随之发生变化,所以结点 i 标签的变化就会导致集合 $c=c_{r(i)}\cup c_{r(j)}$ 中所有结点 f 函数值的变化.下面将集合 c 中的结点分为三类,并针对每一类结点,分别给出其 f 函数值变化的计算公式.

(1)取 $\forall s\in c'_{r(i)}$,其 f 函数值的变化 Δf_s 可被计算为

$$\begin{aligned}\Delta f_s&=f_s(C')-f_s(C)\\&=\sum_{t\in C'(i)}\left(A_{st}-\frac{k_sk_t}{2m}\right)-\sum_{t\in C(i)}\left(A_{st}-\frac{k_sk_t}{2m}\right)\\&=-\left(A_{si}-\frac{k_sk_i}{2m}\right)\end{aligned} \quad (4)$$

(2)取 $\forall p\in c_{r(j)}$,其 f 函数值的变化 Δf_p 可被计算为

$$\Delta f_p = f_p(C') - f_p(C)$$
$$= \sum_{q\in c'_{r(j)}}\left(A_{pq} - \frac{k_p k_q}{2m}\right) - \sum_{q\in c_{r(j)}}\left(A_{pq} - \frac{k_p k_q}{2m}\right)$$
$$= A_{pi} - \frac{k_p k_i}{2m} \tag{5}$$

(3)结点 i 的 f 函数值的变化 Δf_i 可被计算为

$$\Delta f_i = f_i(C') - f_i(C)$$
$$= \sum_{e\in c'_{r(j)}}\left(A_{ie} - \frac{k_i k_e}{2m}\right) - \sum_{e\in c_{r(i)}}\left(A_{ie} - \frac{k_i k_e}{2m}\right) \tag{6}$$

那么,对"由结点 i 标签的变化而导致整个网络的 Q 函数值的变化"ΔQ 之推导如下.

$$\Delta Q = \frac{1}{2m}\left(\sum_{s\in c'_{r(i)}}\Delta f_s + \sum_{p\in c_{r(j)}}\Delta f_p + \Delta f_i\right) \tag{7}$$

$$\Delta Q = \frac{1}{2m}\left(-\sum_{s\in c'_{r(i)}}\left(A_{si} - \frac{k_s k_i}{2m}\right) + \sum_{p\in c_{r(j)}}\left(A_{pi} - \frac{k_p k_i}{2m}\right) + \Delta f_i\right) \tag{8}$$

$$\Delta Q = \frac{1}{2m}\left(-\left(\sum_{s\in c'_{r(i)}}\left(A_{si} - \frac{k_s k_i}{2m}\right) - \left(A_{ii} - \frac{k_i k_i}{2m}\right)\right)\right.$$
$$\left. + \left(\sum_{p\in c_{r(j)}}\left(A_{pi} - \frac{k_p k_i}{2m}\right) - \left(A_{ii} - \frac{k_i k_i}{2m}\right)\right)\right) \tag{9}$$

$$\Delta Q = \frac{1}{2m}\left(\sum_{p\in c'_{r(j)}}\left(A_{pi} - \frac{k_p k_i}{2m}\right) - \sum_{s\in c_{r(i)}}\left(A_{si} - \frac{k_s k_i}{2m}\right) + \Delta f_i\right) \tag{10}$$

$$\Delta Q = \frac{1}{2m}(\Delta f_i + \Delta f_i) = \frac{1}{m}\Delta f_i \tag{11}$$

即 $Q(C') - Q(C) = \frac{1}{m}(f_i(C') - f_i(C))$. 由此可知,若 $f_i(C') > f_i(C)$,则 $Q(C') > Q(C)$,故定理成立.

由性质 1 可知,网络中的每个结点对其 f 函数的计算都仅需要用到局部信息(该结点所在簇的信息);由定理 1 可知,如果网络中某结点标签的变化使得其 f 函数值变大(在其他结点标签不变的前提下),那么该变化将会导致 Q 函数值的增大.在上述理论基础上,我们设计了一个基于局部优化的快速网络聚类算法.不同于传统的从全局观点优化 Q 函数的方法,该算法从局部观点出发,通过使网络中每个结点(利用局部信息)最大化自身的 f 函数,从而达到了优化 Q 函数的目的.

2.3 算法描述

在上节讨论的基础上,我们给出一个基本算法框架如下:

```
Procedure B_FNCA
Begin
1   Assign each node a unique label;
2   Do
3     For each node_i in the network
4         C ← Attain current community structure of the network;
5         labels ← Get current unique labels in the network;
6         Find label_j in labels which can maximize function f of node_i;
7         Assign label_j to node_i;
8     End
9   Until labels of all nodes in the network don't change
End
```

初始时,该算法将为网络中的每个结点定义一个簇;每次迭代中,每个结点都独立计算他取任一标签时的 f 函数值,并选择使其 f 函数值最大的标签作为自身标签;当网络中所有结点的标签都不再发生变化时,算法结束.由于算法第 7 步中每个结点对自身标签的更新都会导致网络簇结构的变化,而第 6 步中每个结点计算其 f 函数时都需要用到网络簇结构信息,所以该算法通过第 4 步来获取(或更新)当前的网络簇结构.

本文的目的有三:提出一个快速的网络聚类算法;该算法仅利用网络的局部信息对 Q 函数进行优化;且可以获得高质量的聚类结果.然而在算法 B_FNCA 中,每个结点计算其 f 函数值并更新自身标签时都需要使用整个网络的簇结构信息(第 6 步),每个结点完成标签更新后都需要重新获取当前网络簇结构(第 4 步),且这两步的时间复杂度都比较高(分析详见 2.4 节).此外,作为一个贪婪搜索算法,该算法容易陷入局部最优解.所以这个初步的算法 B_FNCA 不能很好的满足上述三个要求.

本文通过如下思想来解决这些问题:

在具有簇结构的复杂网络中普遍存在如下直观现象:"网络中任一结点都会与他的某些邻居结点位于同一簇内,或其自身形成一个簇;而出现其他情况是不合理的."因此,在算法第 6 步中,每个结点不需要对当前所有标签来计算对应的 f 函数值,而只需针对所有邻居结点的标签计算其 f 函数值.这样做不但在优化 Q 函数的过程中仅仅使用了网络的部分簇结构信息,而且也显著降低了算法的时间复杂度(分析详见 2.4 节).

此外,我们将算法的第 4 步提到循环外面,即并不是在每一结点完成更新标签操作后都重新计算网络簇结构(同步的标签更新机制),而是等网络中所有结点都完成一次更新标签操作后再重新计算网络簇结构(异步的标签更新机制).这样做不仅提高了算法的效率,而且为算法加入了一种自适应的随机搜索机制.在算法运行初期,由于每个结点更新自身标签的机会都很大,使得网络簇结构变化较大,算法的随机性较强,可以有利于跳出局部最优解;而在算法运行后期,由于每个结点更新自身标签的概率都很小,网络簇结构的变化明显趋缓,该算法几乎又变成了贪婪搜索算法,可以有利于找到更精确的全局最优解.可以看出,这一异步的结点标签更新机制和模拟退火算法[11]中的退火机制有些类似,他可以使网络聚类质量得到进一步改善.

现在我们给出基于 f 函数的快速网络聚类算法 FNCA 如下:

```
Procedure FNCA
Begin
1   Assign each node a unique label;
2   For t = 1: T
3     C ← Get current community structure of the network;
4     X ← Arrange the nodes in the network in a random order;
5     For each node_i ∈ X
6       If the label of any a neighbor of node_i changed at the previous iter-
          ation
7         neighbors_i ← Get neighbors of node_i;
8         neighbors_i ← neighbors_i ∪ node_i;
9         labels ← Get unique labels from neighbors_i;
10          Find label_j in labels which can maximize function f of node_i;
11          Assign label_j to node_i;
12      End
13    End
14    If labels of all nodes in the network don't change
15      Break;
16    End
17  End
End
```

为了提高效率,在算法 FNCA 中,我们通过第 6 步给出了一个加速机制.该机制的主要思想为:“若在上一代中,某结点所有邻居的标签及他们所对应的类簇都没有发生变化,那么在当前代中,该结点的标签也将不会发生变化”,但这个条件似乎太强了;如果将该强条件放宽成弱条件,即“若在上一代中某结点所有邻居的标签均未发生变化,那么在当前代中该结点标签发生变化的概率会很小”,这应该也是合理的.由此,在算法 FNCA 中我们规定满足该弱条件的结点可进入休眠状态,即不需要再重新选择标签;然而当某休眠结点不满足弱条件时,则应马上被唤醒,重新开始计算 f 函数值以选择自身的类标签.该加速机制不但能明显提高算法效率,而且也不会降低网络聚类质量.

算法第 4 步在每次迭代时都首先对网络中所有结点进行随机排序,以此增强了算法的随机性,有利于使算法跳出局部最优解,从而可以得到更好的网络聚类结果.

算法第 2 步增加了一个迭代次数限制 T 作为其运行结束的辅助判定条件,这是因为将“网络中所有结点的标签不再变化”作为算法结束条件是比较苛刻的.实验表明,对大多数网络,即使是具有上百万个结点百万条边的大规模网络,算法迭代 50 次的聚类结果就已足够好了,所以本文所有实验中都将迭代次数限制 T 设定为 50.此外,用户也可以给定一个可以接受的 Q 函数值作为算法终止的辅助判定条件.一般来说,如果 Q 值大于 0.3,则表示网络簇结构明显[9].

可以看出,在算法 FNCA 中,每个结点可以仅利用网络的局部簇结构信息,异步独立的计算其 f 函数以更新自身的类标签,从而达到优化 Q 函数的目的.还应指出的是,FNCA 可被扩展为完全分布式算法,从而实现对真实分布式环境中大规模网络的聚类.

2.4　时间复杂度分析

不妨设算法最大迭代次数限制为 T.假设网络 N 中总的结点数为 n,总的边数为 m,所有结点的平均度为 k,网络最终聚类结果的平均簇规模为 c.由于复杂网络一般为稀疏图,为提高效率,文中所有算法都是通过稀疏矩阵(或称为图链表)来实现的.下面我们给出算法的时间复杂度分析.

性质 2　算法 B_FNCA 的时间复杂度为 $O(T*n^2)$.

证明　算法 B_FNCA 中时间复杂度最高的为第 4 步和第 6 步.其中,执行一次第 4 步(即重新计算一次网络簇结构)的时间为 $O(n)$;执行一次第 6 步(即某结点取遍当前所有标签,同时计算对应的 f 函数值)的时间也为 $O(n)$.由于这两步操作都需要被循环执行 $T*n$ 次,所以该算法时间复杂度为 $O(T*n^2)$.

性质 3　算法 FNCA 的最坏时间复杂度为 $O(T*n*k*c)$.

证明　算法 FNCA 中时间复杂度最高的为第 10 步(即某结点取遍其邻居的所有标签,同时计算对应的 f 函数值).如果不考虑加速机制,该算法进行每次迭代都会执行 n 次第 10 步.由于在算法执行过程中,任一结点邻居的标签都有可能存在重合的现象,所以每次迭代中网络所有结点计算 f 函数的总次数都要小于等于 $n*k$ 次;算法中候选解对应的平均簇规模是随着迭代次数的增加而逐渐变大的,而网络最终聚类结果的平均簇规模为 c,所以算法平均计算一次 f 函数的时间要小于等于 c.因此,该算法每次迭代的时间都不会超过 $O(n*k*c)$.算法 FNCA 至多迭代 T 次,所以其时间复杂度不会大于 $O(T*n*k*c)$.如果考虑到算法加速机制的影响,他的运行速度就更快了.

性质 4　对于真实世界中的大规模复杂网络,FNCA 为近线性算法.

证明　复杂网络一般为稀疏图,即 k 为常数,而 T 又为常数,所以算法 FNCA 的时间复杂度可以表示为 $O(n*c)$.又因为真实世界大规模复杂网络中类簇的平均规模一般都要远小于网络规模(即 $c \ll n$)[15],因此 FNCA 对于真实大规模网络的时间复杂度也可以表示为 $O(n)$.

3 实验

为了定量的分析算法 FNCA 的性能,我们利用人工生成网络和真实世界网络对其进行测试,并给出参数分析.算法实验环境为:处理器 Intel(R) Xeon(R) CPU 5130 @ 2.00GHz 2.00GHz,内存 4.00GB,硬盘 160G,操作系统 Microsoft Windows Server 2003.编程环境为 Matlab 7.3.

3.1 计算机生成的网络

2002 年,Newman 等[1]提出了一个用于测试复杂网络聚类算法精度的基准随机网络模型 $RN(a,s,d,z_{out})$.该网络模型的簇结构已知,其中 a 代表网络中类簇的个数,s 代表每个类簇内的结点数目,d 代表每个结点的度,z_{out} 代表每个结点与簇外结点构成的边数.这里我们采用 $RN(4,32,16,z_{out})$ 来测试文中算法 FNCA 的聚类精度,这一实验方法目前已被相关工作所广泛采用,成为测试网络聚类算法精度的基准方法.很显然,随着 z_{out} 的增大,该网络的簇结构越来越模糊,同时也给网络聚类算法带来了越来越大的挑战.当 $z_{out}>8$ 时,即网络簇间的边数大于簇内的边数,该网络被认为不具有簇结构[1].当且仅当网络中的所有真实类簇被全部正确识别、且没有被进一步划分为多个子簇时,我们才认为该随机网络被完全正确的聚类,本文采用这一方法来计算网络聚类精度.

我们将 FNCA 与算法 GN[1]、FN (Fast Newman)[10]、CPM[5]及 FEC[6]进行比较.其中算法 GN、FN 由 Newman 提出,是以 Q 函数作为目标函数的经典网络聚类算法;而算法 CPM、FEC 仅通过启发式规则(而不使用任何目标函数)对网络进行聚类,分别发表于杂志《Nature》和《TKDE》,也是当前非常优秀的网络聚类算法.图 1(a)给出了实验结果,其中 y-轴表示算法的聚类精度,x-轴表示随机网络中的参数 z_{out}.对于每个算法、每个 z_{out},我们都通过聚类 50 个随机网络取平均准确率.可以看出,文中算法 FNCA 的聚类精度要明显高于其他算法,且 z_{out} 越大,即网络簇结构越模糊时,FNCA 所展现出的优势也就越明显.特别的,当 $z_{out}=8$,即网络簇内的边数和簇间的边数相等时,FNCA 的聚类精度仍保持在 97.83%,而此时其他对比算法的聚类精度已比较低了.

计算速度是另一个评价聚类算法性能的重要指标.2.4 节已给出对算法 FNCA 的时间复杂性分析,本节从实验角度出发来评价该算法的运行效率.图 1(b)给出了文中算法实际运行时间随网络规模增大的变化趋势.由于大规模复杂网络中类簇的平均规模一般要远小于网络规模[15],所以实验中采用了随机网络 $RN(a,100,16,5)$ 进行测试.其中,该网络的类簇规模已确定($s=100$),网络簇的个数可由 a 值自动调节,共包括 $100a$ 个网络结点,$800a$ 条网络连接.图 1(b)中,y-轴表示算法的实际运行时间(以秒为单位),x-轴表示网络规模(结点数 + 连接数).图中 A 曲线表示算法 FNCA 的运行时间,A' 曲线表示算法 FNCA 去掉加速机制后的运行时间.可以看出,该算法具有近线性时间复杂度,并且他的加速机制效果非常明显.

3.2 真实世界的网络

一般来说,真实世界网络与计算机生成的网络会具有一些不同的拓扑特性,因此本节我们通过一些被广泛应用的真实网络来进一步测试算法 FNCA 的性能.其中不仅有包含几十个结点的小规模网络,也有包含上百万结点的大规模网络.表 1 给出了对这些真实网络的简单描述.

在 3.1 节所用到的 4 个对比算法中,GN、FN 两算法与文中算法 FNCA 相同,也是以 Q 函数作为目标函数的方法.在此我们针对表 1 中的真实网络,将 FNCA 与算法 GN 和 FN 进行比较.表 2 给出了实验结果(50 次取平均),其中"–"表示算法内存溢出或 24 小时未运行出结果.可以看出,即使对于具有百万个结点百万条边的大规模网络,算法 FNCA 仍能够在 1 小时之内获得好的聚类结果.该算法的运行效率远优于算法 GN 和 FN,同时聚类质量也要好于这两个算法.

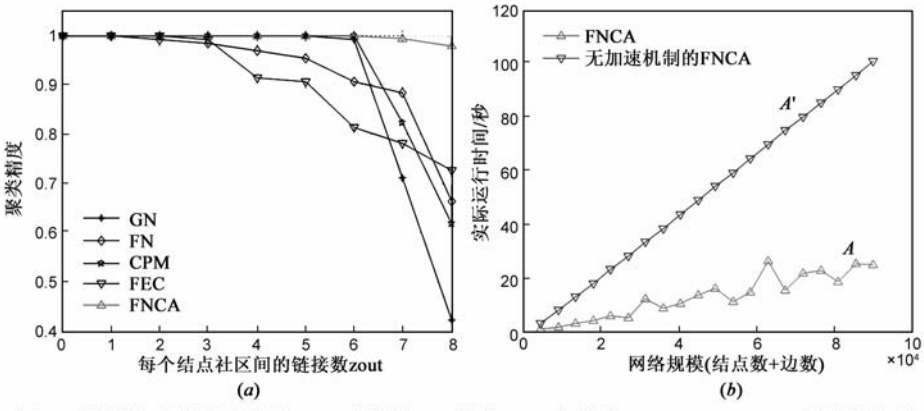


图1 采用随机网络测试算法FNCA的性能;(a) 算法FNCA与算法GN, FN, CPM, FEC聚类精度的比较; (b) 算法FNCA的实际运行时间随网络规模增大的变化趋势

3.3 参数分析

算法 FNCA 几乎是免参数的,因为它只有一个迭代次数限制参数 T,而且还是作为算法运行结束的辅助判定条件.为了阐明 FNCA 的收敛特性,本节以文中四个最大的真实世界网络(WWW, amazon-2003-all, Web-google, Road-PA)作为测试数据对

参数 T 进行了分析.设定迭代次数限制 T 为 50,图 2 给出了算法聚类质量(Q-value)的改善随着其迭代次数增加的变化趋势.很显然,即使对于包含上百万个结点百万条边的大规模网络,算法 FNCA 运行几代就可以得到较好的 Q 值,但要进一步优化 Q 函数则需几十代.但一般来说 T 取 50 已经足够了.

表 1　对实验中所用到的真实网络的简单描述

网络名称	结点数	边数	描述
karate	34	78	Zachary 空手道俱乐部网络[16]
dolphin	62	160	Dolphin 海豚社会网[17]
football	115	613	美国大学足球联赛网络[1]
world	7,207	31,784	语义网络[5]
Cit-hep-th	27,400	352,021	物理论文引用网络[15]
Protein homology	30,727	1,206,654	蛋白质同源网络[18]
arxiv	56,276	315,921	科学家协作网[19]
Epinions	75,877	405,739	epinions.com 域的信任网络[15]
WWW	325,729	1,090,108	nd.edu 域的 WWW 网络[20]
amazon_2003_all	473,315	3,505,519	亚马逊图书网络 2003[15]
Web-google	855,802	4,291,352	谷歌网络 2002[15]
Road-PA	1,087,562	1,541,514	加利福尼亚公路网络[15]

表 2　算法 FNCA 与算法 GN、FN 聚类性能的比较

Q 值/时间(秒)	GN 算法	FN 算法	FNCA 算法
karate	0.4013/0.1015	0.2528/0.031	0.4198/0.0486
dolphin	0.4706/0.3163	0.3715/0.078	0.4969/0.0229
football	0.5996/5.1464	0.4549/0.125	0.6032/0.1635
world	–	0.3821/38.25	0.3920/22.5326
Cit-hep-th	–	0.5189/380.484	0.5875/129.3817
Protein homology	–	0.8612/771.703	0.8923/86.0614
arxiv	–	0.5953/2551.38	0.6165/109.2551
Epinions	–	0.3860/2036.98	0.4151/931.9110
WWW	–	–	0.7190/865.8332
amazon_2003_all	–	–	0.6527/1883.4
Web-google	–	–	0.5931/3391.3
Road-PA	–	–	0.5974/1351.6

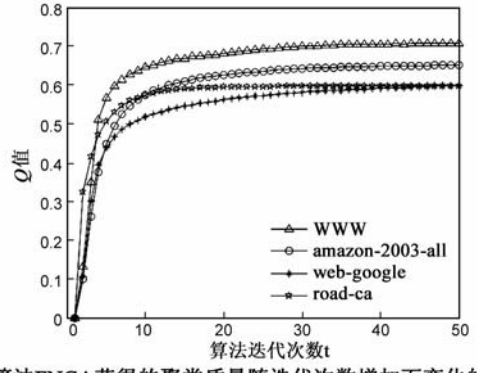


图2　算法FNCA获得的聚类质量随迭代次数增加而变化的趋势

4　结论

本文基于对网络模块性函数 Q 的分析,推导出一个针对于单个结点的局部目标函数 f,并证明 Q 函数随网络中任一结点的 f 函数呈单调递增趋势.基于此,并针对目前复杂网络规模日趋庞大、且呈分布式特性之特点,提出一个基于局部优化的近线性网络聚类算法 FNCA.在 FNCA 中,每个结点仅利用网络的局部簇结构信息来优化自身的目标函数 f,所有结点通过相互协同来实现对整个网络的聚类.实验表明,FNCA 具有速度快、寻优能力强的特点,对于包含百万个结点、百万条边的大规模复杂网络仍能较快的获得高质量聚类结果.

虽然文中算法 FNCA 具有良好的并行特性,但我们并未将其进一步形式化为完全的分布式算法,同时也未进行分布式网络聚类方面的应用;此外,FNCA 中结点的标签更新策略也并不是非常理想,它或许会影响到网络聚类质量.因此,我们希望将以后的工作集中在如下两个方面:(1)试图将 FNCA 扩展为一个完全的分布式算法,并将其应用于真实大规模复杂网络分析(如 P2P 网络、Web 网络等),进而从中发现有重要现实意义的簇结构;(2)通过引入某种自适应的随机机制来完善网络结点的标签更新方法,以期进一步改进算法 FNCA 的聚类性能.

参考文献

[1] Girvan M, et al. Community structure in social and biological networks[J]. Proceedings of National Academy of Science, 2002,9(12):7821 – 7826.

[2] 孔万增,等.基于本征间隙与正交特征向量的自动谱聚类[J].电子学报,2010,38(8):1980 – 1985.
Kong Wan-Zeng, et al. Automatic spectral clustering based on eigengap and orthogonal eigenvector[J]. Acta Electronica Sinica, 2010,38(8):1980 – 1985. (in Chinese)

[3] 杨博,等.复杂网络聚类方法[J].软件学报,2009,20(1):54 – 66.
Yang Bo, et al. Complex network clustering algorithms[J]. Journal of Software, 2009,20(1):54 – 66. (in Chinese)

[4] 王雪松,等.基于复杂网络的时延基因调控网络构建[J].电子学报,2010,38(11):2518 – 2522.
Wang Xue-Song, et al. Construction of delay gene regulatory network based on complex network[J]. Acta Electronica Sinica, 2010,38(11):2518 – 2522. (in Chinese)

[5] Palla G, et al. Uncovering the overlapping community structures of complex networks in nature and society[J]. Nature, 2005, 435(7043):814 – 818.

[6] Yang B, et al. Community mining from signed social networks [J]. IEEE Trans. on Knowledge and Data Engineering, 2007, 19(10):1333 – 1348.

[7] Raghavan U N, et al. Near linear-time algorithm to detect community structures in large-scale networks[J]. Physical Review E, 2007,76(3):036106.

[8] Jin D, et al. Ant colony optimization with Markov random walk for community detection in graphs[A]. Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'11)[C]. Shenzhen, China: Springer-Verlag, 2011.123 – 134.

[9] Newman M E J, et al. Finding and evaluating community structure in networks[J]. Physical Review E, 2004, 69(2): 026113.

[10] Newman M E J. Fast algorithm for detecting community structure in networks[J]. Physical Review E, 2004, 69(6): 066133.

[11] Guimera R, et al. Functional cartography of complex metabolic networks[J]. Nature, 2005, 433(7028): 895 – 900.

[12] Barber M J, et al. Detecting network communities by propagating labels under constraints[J]. Physical Review E, 2009, 80(2): 026129.

[13] Liu X, et al. Advanced modularity-specialized label propagation algorithm for detecting communities in networks[J]. Physica A, 2010, 389(7): 1493 – 1500.

[14] Jin D, et al. Genetic algorithm with local search for community mining in complex networks[A]. Proceedings of the 22th International Conference on Tools with Artificial Intelligence (ICTAI'10)[C]. Arras, France: IEEE Press, 2010.105 – 112.

[15] Leskovec J, et al. Statistical properties of community structure in large social and information networks[A]. Proceedings of the 17th International Conference on World Wide Web (WWW'08)[C]. Beijing, China: ACM Press, 2008. 695 – 704.

[16] Zachary W W. An information flow model for conflict and fission in small groups[J]. J. Anthropological Research, 1977, 33(4): 452 – 473.

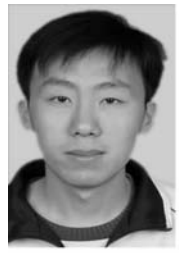
[17] Lusseau D. The emergent properties of a dolphin social network[J]. Proc Biol Sci, 2003, 270(Suppl2): S186 – S188.

[18] University of Texas Bioinformatics, Proteomics, and Functional Genomics server. Similarity between two proteins using the BLAST algorithm[DB/OL]. http://apropos.icmb.utexas.edu/lgl/, 2002.

[19] Newman M E J. The structure of scientific collaboration networks[J]. Proceedings of National Academy of Science. 2001, 98(2): 404 – 409.

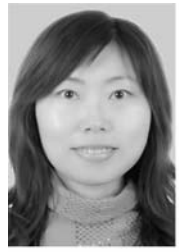
[20] Center for Complex Network Research. Edgeed WWW pages in the nd.edu domain[DB/OL]. http://www.nd.edu/-networks/resources/, 2007.

作者简介



金 弟 男,吉林大学博士研究生.主要研究方向为数据挖掘,复杂网络分析,多 Agent 系统.
E-mail: jindi.jlu@gmail.com

刘大有 男,吉林大学计算机科学与技术学院教授.主要研究方向为知识工程、专家系统与不确定性推理、时空推理、分布式人工智能、多 Agent 和移动 Agent 系统、数据挖掘与多关系数据挖掘、数据结构与计算机算法.
E-mail: dyliu@jlu.edu.cn



刘 杰(通信作者) 女,吉林大学副教授,博士.主要研究方向为数据挖掘,模式识别.
E-mail: liu_jie@jlu.edu.cn