Journal of Computer Applications
计算机应用,2012,32(8):2154-2158
ISSN 1001-9081
CODEN JYIIDU
2012-08-01
http://www.joca.cn

文章编号:1001-9081(2012)08-2154-05
doi:10.3724/SP.J.1087.2012.02154

基于"次中心"的社区结构探寻算法

水　超[1*],李　慧[2]
(1.国防科学技术大学 信息系统与管理学院,长沙 410073;　2.国防科学技术大学 信息中心,长沙 410073)
(*通信作者电子邮箱 super_shuichao@163.com)

摘　要:当前社区结构探测算法在寻求社区结构划分正确性的同时,算法效率较低。为此,提出一种在算法正确性和算法效率两个方面能取得较好均衡的社区结构探寻算法 CoreScan。该算法寻找节点集合中一类称之为"次中心"的特殊节点,再将其作为聚类中心,然后通过 D 模块度来发现社区结构。理论分析表明,该算法能正确识别 Fortunato 提出的一类特殊社区结构,且算法效率可达 $O(n*k_{max})$,其中 n 是节点数量,k_{max} 是"次中心"最大数量。最后通过多项实验证明,CoreScan 算法能够在效率和正确性上取得较好的均衡,适合于在大规模节点集合中进行快速社区结构探寻。

关键词:社区结构;社区探测;Q 模块度;D 模块度
中图分类号:TP18　　**文献标志码**:A

Community structure identification algorithm based on subcenter

SHUI Chao[1*], LI Hui[2]
(1. *College of Information System and Management, National University of Defense Technology, Changsha Hunan* 410073, *China*;
2. *Information Center, National University of Defense Technology, Changsha Hunan* 410073, *China*)

Abstract: Some community structure identification algorithms in complex network research achieve high quality for identifying community correctly but low efficiency, but some algorithms were on the contrary. It is a challenge for community structure identifying algorithms to balance efficiency and correctness for category meanwhile. In this paper, a new algorithm named CoreScan was proposed. Different from existing algorithms, CoreScan found some special node named subcenter in network and using a preprocessing phase to divide nodes into several communities according to subcenter, then cluster nodes by evaluating the D modularity of community. The definition of subcenter was given and some certifications also were shown to guarantee the correction of CoreScan algorithm. At last, the algorithm was tested on two artificial networks and two real-world graphs. The experimental results show that the algorithm achieves the goal of linear efficiency, and the correction of identifying community is no less than existing algorithms. This algorithm is suitable to large-scale network structure investigation.

Key words: community structure; community detection; Q modularity; D modularity

0　引言

社区结构是复杂网络的三个主要特征之一,在大型复杂网络中发现和探测网络结构是当前复杂网络研究的一个重要方向。2002 年,Newman 提出社区结构评价指标 Q 模块度以来,基于模块度的社区结构探测算法研究一直方兴未艾。该类算法主要追求两个目标,一个是提高算法的效率使之可以适应于大规模节点网络,另一个是提高社区探测的正确性。Danon 等[1]在比较了当前的社区探测算法后指出,目前已有的算法在算法效率和算法正确性上成反比关系。因此建立一种新算法,既可保证算法的高效率,同时也保证算法的正确性成为一个挑战性的课题[1-2]。本文根据 Q 模块度和 D 模块度的性质提出了一种 CoreScan 社区算法,其算法效率为 $O(n*k_{max})$,其中 n 是节点数量,k_{max} 是算法获得的最大社区数。一般来说,$k_{max} \ll n$,因此在一般情况下该算法接近线性。CoreScan 算法保持了对大规模社区探测的正确性,且能发现 Fortunato 等[3]提出的小规模网络,较好地实现了保持算法高效和正确探测社区的双重目标。

1　相关研究

在社会活动中,人们往往根据各种关系形成不同的社区,如校园中的足球俱乐部、诗歌俱乐部、戏曲俱乐部等。如果将社区中的人看成网络中的一个节点,将社区成员的联系看成是网络中的连线,那么一个社区可以用一个小型网络来表示,称之为"社区网络"。多个社区网络连接在一起,就形成了一个复杂网络。社区探测算法的任务就是从复杂网络中重新将这些社区网络区分离开。为统一描述,本文设 V 代表复杂网络,m 为 V 的边数,n 为 V 的节点数,k_i 代表节点 i 的度数,C_i 代表算法获得的第 i 个社区,k_{max} 代表算法获得的最大社区数,$\boldsymbol{A}$ 为网络的连接矩阵,即 $\boldsymbol{A}=\{a_{ij} \mid a_{ij}=1$,当节点 i 与节点 j 有连接,否则 $a_{ij}=0\}$。因此,Newman 提出的 Q 模块度可以表示如下:

$$Q=\frac{1}{2m}\sum\left(A_{ij}-\frac{k_ik_j}{2m}\right)\delta(C_i,C_j)$$

该模块度的物理含义是社区内部连接密度大于社区外部连接密度。它可较好地评价探测算法的质量,因而成为人们关注的热点。Newman 等[4]提出的 GN 算法是第一个基于 Q

收稿日期:2012-02-23;**修回日期**:2012-03-23。
作者简介:水超(1976-),男,湖南长沙人,助理研究员,博士,主要研究方向:科学计量、复杂网络;　李慧(1976-),女,湖南醴陵人,工程师,硕士,主要研究方向:多媒体应用。

模块度的社区探测算法,但其算法的效率最差复杂度为 $O(n^5)$,在稀疏网络中的复杂度为 $O(n^3)$,不能适应于大规模网络。其后,基于模块度的社区探测算法的一个重要研究方向就是提高算法效率以适应于大规模网络,如 Newman[5] 提出的贪婪算法(NG 算法),算法效率为 $O((m+n)n)$; Duch 等[6] 提出的 DA 算法,算法效率在通过堆优化后可达到 $O(n^2 \log n)$; Clause[7] 利用堆结构对 Newman 等的贪婪算法进行改进,提出了 CNM 算法,其算法效率接近线性复杂度,为 $O(n\log^2 n)$; Wu 等[8] 提出的 WH 算法是第一个达到线性的社区探测算法,算法效率为线性 $O(m+n)$,但是该算法需要预先知道社区的数量,并假定了社区节点数量的差值不超过预先设定的阈值 β,限制了该算法的使用范围;Duch 等[6] 提出的基于极值的 EQ 算法,其算法效率为 $O(n^2 \log n)$; Fortunato 等[9] 提出的 FLM 算法,其算法效率为 $O(n^4)$; 赵凤霞等[10] 提出了一种类似于 K-Means 方法的社区探测算法,并认为该算法的效率为 $O(n)$,但笔者认为该算法实际上的效率应该是 $O(kmn)$; 此外的算法还包括 Radicchi 等[11] 提出的边聚集系数算法(RCCLP),Newman[12-13] 提出来的具有 Q 模块度的谱分析算法等。

算法研究的另一方向是提高探测社区算法的正确性。目前,人们主要采用 Newman 构造的人工网络、实际网络中的 karake Club 社区、American football college 社区、物理家合作网络等真实社区作为检验社区探测算法正确性的基准。各类算法对这些网络探测的正确性成为研究追求的目标。

2007 年,Fortunato 等[3] 通过数学分析指出,当一个社区内部连线数小于 $\sqrt{m/2}$ 后,即使代表该社区的节点形成了一个完全图且与外部社区连线数为 1,也无法通过优化 D 模块度的算法将其识别出来。

同样,在加权网络中当社区内部连线数小于 $\sqrt{W\varepsilon/2}$ 时(W 是网络的全部权重,而 ε 是社区内最大权重),该社区也不能被正确地探测[14]。Good 等[15] 进一步分析后指出这种限制不仅和网络的边数有关,而且与网络中包含的社区数有关,基于模块度优化的社区探测算法都不可避免地存在着这个问题。

为此,文献[16] 提出了 D 模块度的概念,一个社区的 D 模块度为该社区内部连线数与外部连线数的差值与社区节点数量的比值,即 $D = \sum_{i=1}^{l} d(C_i) = \sum_{i=1}^{l} \frac{L(C_i,C_i) - L(C_i,\bar{C}_j)}{|C_i|}$。该文中证明了优化 D 模块度不会导致小规模社区被大社区吞噬的情况。文献[16] 采用了类似于 K-means 的算法来验证自己的结果,该算法同样需要事先知道社区的数量。

在文献[16] 工作的基础上,Blondel 等[17] 提出了一种基于 D 模块度增益的算法(VD 算法),算法的效率为 $O(kmn)$,其中 k 是算法迭代的最大次数。但是该算法只是在一定程度上避免了 Fortunato 设想的情况,效果并不理想。Berry 等[14] 通过对网络进行赋权的方法,使得加权网络中小规模社区的最大权重尽可能地小,从而使得 D 模块度增益算法能发现小规模社区。该算法根据 CNM 算法进行改进,因此算法效率也是 $O(n\log^2 n)$。因此,当前基于模块度的社区探测算法需要面对几个方面的问题:1)需要高效的算法以适应于大规模网络;2)正确发现社区;3)算法应该是无监督的,即无需事先知道社区数量。

2　社区发现算法

2.1　算法基本思想

首先讨论 D 模块度和 Q 模块度的物理意义。从 D 模块度的定义来看,它代表了是社区内部连线密度与社区外连线密度的差值。当 D 模块度为正值时,有 $L(C_i,C_i) > L(C_i,\bar{C}_i)$;但是当社区内部节点数较小时,$D$ 模块度为负值的概率较大。这就意味着只有社区规模达到一定程度后,D 模块度的增益才有意义。Q 模块度则代表了社区内部连接密度大于社区外部连接密度。我们可以证明与 Q 模块度相关的两个推论。

推论 1　在 Q 模块度全局最大时的社区划分算法过程中,如果两个节点彼此不连接,则它们度数越高,其在同一社区的概率越小。

证明　在以 Q 模块度最大化为目标的算法中,设 L 为算法过程中的一个中间状态,它将节点集合划分为 k 个社区,社区 C_a 是该划分中的任意一个社区,则 a 在下一个中间状态继续为社区的概率为:

$$\rho = Q_a/Q$$

其中 Q_a 为社区 a 的 Q 模块度,且

$$Q_a = \frac{1}{2m}\sum\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(C_{ia},C_{ja}) = \frac{1}{2m}\left(\sum^{l}\left(1 - \frac{k_i k_j}{2m}\right) + \sum\left(-\frac{k_i k_j}{2m}\right)\right)$$

因此,社区 a 继续成为社区的概率取决于两个方面,一个是社区中连线的密度,一个是社区中不相连节点的度数乘积。当社区 C_a 中节点 i 和节点 j 不连接时,它们对 Q_a 的增益是 $-k_i k_j/2m$,i 和 j 的度数越高,则负增益越大,即社区 C_a 在下一个中间状态成为社区的概率越小。所以节点 i 与节点 j 在一个社区的概率越小。

同理,可以证明第二个推论。

推论 2　与大度数节点相连接的节点,度数越小则与之在同一个社区的概率越大。

这与社会生活中人们观察到的情况相一致。在实际生活中,一个社区虽然可能有大量人员参加,但是其中最活跃的人物却不多。这些活跃分子与社区中大部分人联系,并组织和完成社区中的大部分活动,因此可称他们为“核心人物”。在同一社区,两个核心人物之间不交往的概率较小,而同一社区中其他人与这些核心人物进行交往的概率较高。

利用上述两个推论,算法的基本思想是寻找节点集合中互不相连且节点度数尽量高的节点,并将之分配到不同的社区,使得社区以较高的概率达到 Q 模块度最大。为此,本文提出了“次中心节点”的概念,它是社区中度数最高且不与外部高于自身度数的次中心节点相连的节点,用 $DCentre(C_i)$ 表示:

$$DCentre(C_i) = \max Degree(\{b_k \mid b_k \in C_i\}),$$

且对于 $\forall d_j = DCentre(C_j)$ 有 $a_{b_k d_j} = 0$,

且 $Degree(b_k) \geqslant Degree(d_j)$

根据次中心节点的定义,可获得推论 3。

推论 3　存在着划分 L,使得节点集合 V 划分为 $k_{\max}$ 个社区,$k_{\max}$ 是节点集合 V 中次中心节点的数量,每个社区中有且只有一个次中心,且每个节点唯一分配到一个社区中。

证明　根据下列过程,可将节点集合划分为 $k_{\max}$ 个社区:

1) 取出节点集合 V 中度数最高的节点,设该节点为核心节点;

2) 将节点集合 V 中所有与该节点相连的节点与该核心节点合并为一个社区 C,且 $V = V - C$;

3) 重复步骤 1)、2) 直至 $V = \varnothing$。

该过程将节点集合 V 划分为 k 个社区,且每个社区有一个核心节点,每个节点仅属于一个社区,下面还需证明核心节点即次中心节点,且 $k = k_{max}$。

假设节点 i 是一个核心节点,且 $V_i \in C_i$。根据步骤 1),$D(V_i) = \max\left(V - \sum C_k\right)$,而根据划分步骤 1)、2) 有 $D(V_{i-1}) < D(V_i)$,且 C_i 中的节点不与 C_{i-1} 中的节点连接,所以节点 i 是次中心节点。

假设节点 j 是社区 C_i 的另外一个次中心节点,则根据步骤2),V_i 与 V_j 相连接,与次中心节点定义矛盾。所以 C_i 不存在两个次中心节点。

2.2 算法概述与分析

根据上述思想,我们提出了 CoreScan 算法,具体如下:

1) 根据推论 3 证明过程来寻找网络中的次中心节点,然后计算其他节点与相连次中心点的余弦相似度,将该节点唯一分配到相似度最大的社区内,从而形成了 k_{max} 个社区。

2) 寻找包含次中心节点的数量为 k 的完全子图,以确定各个社区中的核心节点群 $Core(C_i, r)$,实际算法中取 $r = 3$。

3) 将其他非核心节点从社区删除,再计算某个节点 n_j 与社区 C_i 核心节点群的余弦相似度,即

$$Similar = \sum Consize(j,k)/r;\ N_k \in Core(C_i, 3)$$

并将该节点加入相似度最大的社区。

4) 将各个社区按照社区相互之间的 D 模块度进行层次化聚合,即根据 D 值增益最大以合并社区 C_i 和 C_j。

CoreScan 算法的伪代码如下:

```
初始化:
  社区集合 C = ∅
  邻接矩阵 A = {a_ij},且 a_ii = degree(n_i)
  节点集合 ref = {n_i} 且 degree(n_i) ≥ degree(n_j)(i > j)
1)  for each n_i ∈ ref do
      If ((a_{DCentre(C_i),n_i} = 1) && degree(n_i, DCentre(C_j)) =
        max{DCentre(C_j), n_k})
      do { C_j = C_j ∪ n_i
        ref = ref − n_i }
      else{
        create new community C_k = {n_i}
        and DCentre(C_k) = n_i}
    If (ref ≠ ∅) goto 2)
2)  计算每个社区的核心三角 CoreTriangle
3)  For each b ∉ CoreTriangle do{
      C_i = C_i − b; ref = ref + b;}
    for each b ∈ ref do{
      similar_i(b, C_i) = computerQ;
      similar_x = max(similar_i);
      C_x = C_x ∪ b
      ref = ref − b;}
4)  for each C_i do {
      for each C_j do{
        ΔD = max(D(C_i + C_j) − (D(C_i) + D(C_j)))
        D_xy = max(ΔD)}
    Merge(C_x, C_y);}
```

在 3.1 节中将采用空手道俱乐部的例子进行算法过程说明。

算法的特点在于:根据推论 3 中提出的划分方法将所有节点划分到 k_{max} 个社区后,可使得社区达到一定规模,再使用 D 模块度进行社区合并,避免了 D 模块度不适合小规模节点的情况;当社区规模较小时,要到达社区内部连线多于外部连线的要求,其与外部连线就越少,则出现次中心节点的概率越高,从而有利于发现图中的小规模节点,避免了 Q 模块度不能发现小规模节点的情况。此外,相比以前的算法,CoreScan 取得了较好的效率:算法在 1)、2) 步各需要计算 $k_{max} * n$ 次,在第3) 步需要计算 $(k_{max} + 1) * n$ 次,在计算第4) 步时最多需要 $k_{max}^2/2$ 次,则整个算法需要 $4 * k_{max} * n + k_{max}^2/2$ 步,即算法效率为 $O(k_{max} * n + k_{max}^2)$。一般来说,$k_{max}$ 是远远小于 n 的,只有在完全非连通图时才有 $k_{max} = n$ 的情况。因此相对当前已有的算法,CoreScan 算法的效率是线性,从而极大提高了算法效率。

3 实验与结果分析

3.1 空手道俱乐部数据集实验

20 世纪 70 年代初,Wayne Zachary 观察了美国大学空手道俱乐部(Karake)成员之间的人际关系,并根据成员间交往状况建立了一个网络。这个网络包含 34 个顶点,代表了俱乐部成员;有 78 条边,代表俱乐部成员之间的人际关系。

本章用 CoreScan 算法分析该网络,并详细说明算法执行的动态过程,其社区聚类过程如下所示:

1)算法识别节点集合中的次中心节点。首先节点 1 的度数最高,它作为第一个次中心点。不与节点 1 相连接,且度数最高的是节点 34,它作为第二个次中心点。如此迭代,算法将该网络分为了 5 个社区,括号中的编号代表了算法发现的次中心节点,每一行代表了与次中心相连且根据余弦相似度聚类到该社区的节点。

①(1),22,20,18,14,13,12,5,4,8

②(34),10,30,29, 27,24,28

③(33),23,21,19,16,15,9,31,3

④(26),25

⑤(17)

2)算法在每个社区中分离出社区的核心节点。括号内是该社区的核心节点编号,其余为根据余弦相似度聚类到该社区的节点编号。

①(4,8,1),22,20,18,14,13,12, 5, 3,1

②(24,28,34)10,30,29, 27

③(9,31,33), 23,21,19,16,15

④(26),32,25

⑤(17),11,7,6

3)算法根据各个社区按照社区相互之间的 D 模块度进行层次化聚合,合并过程如下,最终合并结果如图 1 所示。

①第 2 社区与第 3 社区合并为第 7 社区

②第 4 社区与第 7 社区合并为第 8 社区

③第 1 社区与第 5 社区合并为第 9 社区

④第 9 社区与第 8 社区合并为第 10 社区

在 Karake Club 网络中,3 号节点的划分一直存在争议,这是因为该节点与左右 2 个社区的连线都是 5(如图 1 所示),因此无法区分该节点归属。当 CoreScan 算法对 karake club 社区进行划分中,左右两个社区还可以划分为 5 个社区,其中左边可以分裂为 3 个,而右边可以分裂为 2 个。3 号节点此时就与以 4、8、1 号为核心的社区联系更加紧密,这也就解释了为什么 3 号节点会在该社区。算法合并社区后,在社

区数量为 4 时,到达 D 模块度最高值,如图 1 所示,虚线将图划分成了 4 个社区。当社区继续合并成 2 个社区时,形成的社区与实际情况完全一致。

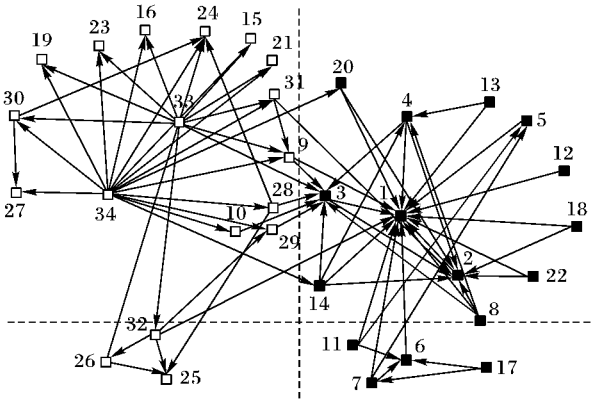

图 1　算法将 Karake 分为 4 个社区

3.2　人工网络数据集实验

人工网络是当前复杂网络社区划分的一个基准。它由 128 个节点组成,分为 4 个社区,每个社区有 32 个节点。每个节点与社区内节点有 Z_{in} 条连线,而与社区外节点有 Z_{out} 条连线,且 $Z_{in}+Z_{out}=16$。大部分算法都以该网络为基准进行实验。设节点分配正确率为:正确分配的节点数/全部节点数,则目前绝大多数实验表明在 $Z_{out}<8$ 时,大部分算法的正确率都较高,而到 $Z_{out}\geqslant 8$ 时,大部分算法的正确率快速下降。CoreScan 算法在该人工网络实验中,当 $Z_{out}=6$ 时,正确率为 92.1%;$Z_{out}=7$ 时,正确率为 81.7%;$Z_{out}=8$ 时,正确率为 69.4%。根据 Danon 等[1]收集的情况,只有 DA 算法、SA 算法、LP 算法的正确率超过了 CoreScan 算法,但这些算法的效率为 $O(n^2)$ 或以上。

在第二种人工网络中,本文采用了 Fortunato 构造的一种特殊的社区结构。实验表明 CoreScan 算法对该社区识别正确率为 100%,且未发生小规模社区被大规模社区吞并的情况。

由于 Fortunato 社区人工网络不具有一般性,因此我们构造了一种新的人工网络。该网络首先构造 20 个节点数为 5 的完全子图,每个子图与两个完全子图相连,形成一个与 Fortunato 社区相同的环结构;然后在每个完全子图中随机选择 r 个节点,每个节点以概率 $\rho=0.5$ 与完全子图外的节点相连,连线数量为 β。图 2 展示了当 $r=1$,$\beta=1$ 时形成的一个社区结构。

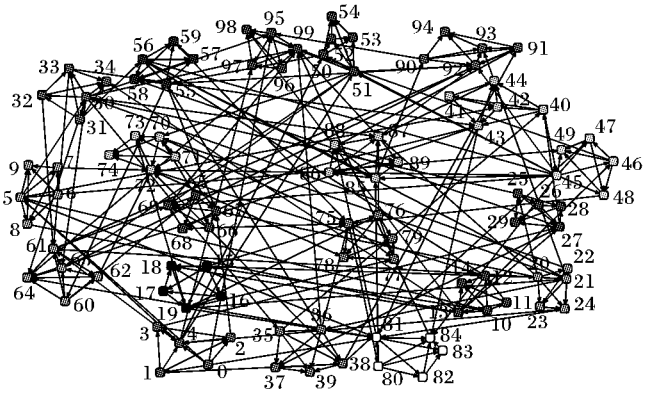

图 2　20 个节点数为 5 的完全子图

实验首先让 $\beta=1$ 且 r 从 1 增长到 5,分析算法与外连节点的关系,在 10 次实验中 CoreScan 算法全部都正确获得社区结构。实验设置 $r=1$,将 β 从 1 增长到 20,则正确识别社区的数量如图 3 所示。GN 算法由于 Q 模块度的限制而只能在 $\beta=1$ 时发现 10 个社区,CoreScan 算法则能发现全部 20 个社区;当 $\beta=20$ 时,网络已经接近随机网络,此时 GN 算法只能发现 6 个社区,而 CoreScan 算法可以发现 13 个社区。

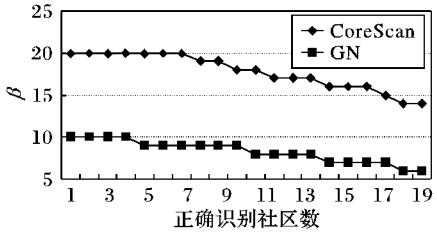

图 3　不同算法发现社区的个数

3.3　美国大学生橄榄球联盟数据集实验

Newman 考察了美国大学生橄榄球联盟,并收集了该联盟 2000 年的真实比赛情况。该联盟有 12 个子联盟,并可以用一个节点数量为 115,边数为 616 的图表示该联盟的比赛情况,其中节点代表了联盟中的球队,边代表了球队之间的比赛。通过 CoreScan 算法,当社区合并为 11 个时,D 模块度达到最大值。具体情况如表 1 所示,其中括号中的数字代表该节点真实所处的社区序号。

表 1　美国大学生橄榄球联盟数据集实验结果

社区	节点
1	40(3),84(3),107(3),81(3),102(3),10(3),52(3),98(3),55(3),74(3),3(3),72(3),2(2)
2	41(7),16(7),104(7),9(7),4(7),93(7),23(7),0(7)
3	64(2),39(2),15(2),60(2),106(2),13(2),32(2),100(2),6(2),47(2)
4	114(11),67(11),88(11),110(4),83(11),53(11),73(11),49(11),46(11)
5	422(8),68(8),21(8),111(8),108(8),78(8),77(8),8(8),7(8),51(8)
6	113(9),20(9),65(9),87(9),17(9),62(9),56(9),76(9),27(9),96(9),95(9),70(9)
7	43(6),42(5),18(6),85(6),38(6),61(6),14(6),12(6),34(6),54(6),31(6),99(6),26(6),71(6)
8	19(1),82(5),35(1),80(5),79(1),101(1),55(1),30(1),29(1),94(1)
9	45(0),89(0),109(0),37(0),105(0),103(0),33(0),25(0),1(0)
10	91(4),44(4),112(4),66(4),63(10),86(4),36(5),59(10),58(11),57(4),75(4),97(10),48(4),92(4)
11	50(10),28(11),69(10),90(5),24(10),11(10)

从表 1 中可以看出,没有被正确识别的社区是第 5 号社区,即 IA Independents 社区,该结果与 GN 算法相同。这是因为该社区内部比赛数量小于外部比赛数量,导致了社区识别失败。此外,与 GN 算法相同的是第 10 号社区即“sunbelt”社区,在本算法中也被分为了两个社区,其中一个被 coferneceUSA 社区合并,预示该社区可能被分裂。

GN 算法将橄榄球联盟分为 13 个社区和 4 个孤立点,而贪心算法则只发现了 6 个社区。CoreScan 算法的结果中有 11 个球队没有被正确划分,正确率为 90.1%,与 GN 算法正确率相当,而高于贪心算法 78% 的正确率。CoreScan 算法的 Q 模块度值为 0.600 51,高于 NG 算法的 0.589,而低于 GN 算法的 0.601。

4 结语

CoreScan 算法将网络划分为以次中心节点为核心的 k_{max} 个社区，并将其他节点唯一分配到该社区，从而减小了初始合并社区的数量，使得算法的效率接近线性。实验表明在实际社会网络中，该算法取得了良好的效果，但不适合随机网络。该实验同时说明在实际的社会生活中，一些节点并不与社区外部连接，它们构成的次中心节点存在并影响了社区的结构。如何更好地利用这些次中心节点，以提高算法的正确性是我们下一步的工作。

参考文献：

[1] DANON L, DíAZ-GUILERA A, DUCH J, *et al*. Comparing community structure identification [J]. Journal of Statistical Mechanics: Theory and Experiment, 2005, 2005(9): 09008 – 09018.

[2] FORTUNATO S. Community detection in graphs [J]. Physics Reports, 2010, 486(3): 75 – 174.

[3] FORTUNATO S, BARTHELEMY M. Resolution limit in community detection [J]. Proceedings of the National Academy of Sciences, 2007, 104(1): 36 – 41.

[4] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. Proceedings of the National Academy of Sciences, 2002, 99(2): 7821 – 7826.

[5] NEWMAN M E J. Fast algorithm for detection community structure in networks [J]. Physical Review E, 2004, 69(6): 066133.

[6] DUCH J, ARENAS A. Community detection in complex networks using extremal optimization [J]. Physical Review E, 2005, 72(2): 027104.

[7] CLAUSET A. Finding local community structure in networks[J]. Physical Review E, 2005, 72(2): 026132.

[8] WU F, HUBERMAN B A. Finding communities in linear time: a physics approach [J]. The European Physical Journal B: Condensed Matter Physics, 2004, 38(2): 331 – 338.

[9] FORTUNATO S, LATORA V, MARCHIORI M. Method to find community structures based on information centrality [J]. Physical Review E, 2004, 70(5): 056104.

[10] 赵凤霞，谢福鼎. 基于 K-means 聚类算法的复杂网络社团发现新方法[J]. 计算机应用研究, 2009, 26(6): 2041 – 2049.

[11] RADICCHI F, CASTELLANO C, CECCONI F, *et al*. Defining and identifying communities in networks [J]. Proceedings of the National Academy of Sciences, 2004, 101(9): 2658 – 2663.

[12] NEWMAN M E J. Modularity and community structure in networks [J]. Proceedings of the National Academy of Sciences, 2006, 103(23): 8577 – 8582.

[13] NEWMAN M E J. Finding community structure in networks using the eigenvectors of matrices [J]. Physical Review E, 2006, 74(3): 036104.

[14] BERRY J W, HENDRICKSON B, LAVIOLETTE R A, *et al*. Tolerating the community detection resolution limit with edge weighting [J]. Physical Review E, 2011, 83(5): 056119.

[15] GOOD B H, de MONTJOYE Y-A, CLAUSET A. The performance of modularity maximization in practical contexts [J]. Physical Review E, 2011, 81(4): 046106.

[16] LI ZHENPING, ZHANG SHIHUA, WANG RUI-SHENG, *et al*. Quantitative function for community detection [J]. Physical Review E, 2008, 77(3): 036109.

[17] BLONDEL V D, GUILLAUME J-L, LAMBIOTTE R, *et al*. Fast unfolding of communities in large networks [J]. Journal of Statistical Mechanics: Theory and Experiment, 2008, 2008(10): 10008 – 10014.

（上接第 2153 页）

要其中的任意 70% 的节点，即可容忍集群中不超过 30% 的节点同时失效，提高了数据的安全性。

4 结语

基于 GE 码和动态副本策略的方案主要是针对 HDFS 存储效率不高以及负载均衡能力不足而提出来的。数据实验测试结果表明，Noah 在保证了集群数据安全性的同时，提高了数据恢复的速度，降低了整体的存储成本，优化了 HDFS 的负载均衡能力。

目前该方案遇到的困难主要集中在 section 的管理上，随着集群规模的扩大，section 数量也迅速增加，文件映射表的建立和维护成本也随之增加，从而导致集群的扩展性受到了制约，在今后的工作当中，将进一步研究如何降低所引入的系统复杂性以及如何设计更加优化的 section 分发策略，逐步完善改进的系统方案。

参考文献：

[1] GHEMAWAT S, GOBIOFF H, LEUNG S-T. The Google file system [C]// SOSP 2003: Proceedings of 19th ACM Symposium on Operating Systems Principles. New York: ACM, 2003: 58 – 66.

[2] Hadoop[EB/OL]. [2011 – 12 – 01]. http://hadoop.apache.org/common.

[3] Map-Reduce[EB/OL]. [2011 – 12 – 01]. http://wiki.apache.org/mapreduce/.

[4] BORTHAKUR D. Hadoop and its usage at Facebook [R/OL]. [2011 – 11 – 13]. http://borthakur.com/ftp/newflixalone.pdf.

[5] MACKEY G, SEHRISH S, WANG JUN. Improving metadata management for small files in HDFS [C]// CLUSTER '09: IEEE International Conference on Cluster Computing and Workshops. Piscataway: IEEE, 2009: 1 – 4.

[6] THUSOO A, SARMA J S, JAIN N, *et al*. Hive: A petabyte scale data warehouse using Hadoop[C]// ICDE 2010: Proceedings of 26th IEEE International Conference on Data Engineering. Piscataway: IEEE, 2010: 996 – 1005.

[7] JIANG D, OOI B C, SHI L, *et al*. The performance of MapReduce: an in-depth study[J]. Proceedings of the VLDB Endowment, 2010, 3(1/2): 472 – 483.

[8] DONG BO, QIU JIE, ZHENG QINGHUA, *et al*. A novel approach to improving the efficiency of storing and accessing small files on Hadoop: a case study by PowerPoint files[C]// SCC '10: Proceedings of the 2010 IEEE International Conference on Services Computing. Washington, DC: IEEE Computer Society, 2010: 65 – 72.

[9] LIU XUHUI, HAN JIZHONG, ZHONG YUNQIN, *et al*. Implementing WebGIS on Hadoop: a case study of improving small file I/O performance on HDFS[C]// CLUSTER '09: IEEE International Conference on Cluster Computing and Workshops. Piscataway: IEEE, 2009: 45 – 48.

[10] HDFS [EB/OL]. [2011 – 12 – 05]. http://hadoop.apache.org/common/.

[11] NameNode[EB/OL]. [2011 – 12 – 11]. http://wiki.apache.org/namenode/.

[12] DataNode[EB/OL]. [2011 – 12 – 21]. http://wiki.apache.org/datanode/.

[13] 陈铮. 一类新的阵列纠删码理论及应用[D]. 北京：中国科学院研究生院, 2009.

[14] BlocksMap[EB/OL]. [2011 – 02 – 08]. http://www.tbdata.org/archives/1120/.

[15] WHITE T. Hadoop: The definitive guide[M]. Sebastopol, California: O'Reilly Media, 2011: 295.