基于结构相似度的大规模社交网络聚类算法

陈季梦[①] 陈佳俊[②] 刘 杰*[①] 黄亚楼[②] 王 媛[①] 冯 霞[③]
[①](南开大学计算机与控制工程学院 天津 300071)
[②](南开大学软件学院 天津 300071)
[③](中国民航大学民航信息技术科研基地 天津 300300)

摘 要：针对社交网络的有向交互性和大规模特性，该文提出一种基于结构相似度的有向网络聚类算法(DirSCAN)，以及相应的分布式并行算法(PDirSCAN)。考虑社交网络中节点间的有向交互性，将行为结构相似的节点聚集起来，并进行节点功能分析。针对社交网络规模巨大的特点，提出MapReduce框架下的分布式并行聚类算法，在确保聚类结果一致的前提下，提高处理性能。大量真实数据集上的实验结果表明，DirSCAN比无向网络聚类算法(SCAN)在F1上可提高2.34%的性能，并行算法PDirSCAN比DirSCAN运行速度提升1.67倍，能够有效处理大规模的有向网络聚类问题。

关键词：社交网络；有向网络聚类；并行算法；MapReduce
中图分类号： TP393 **文献标识码**： A **文章编号**：1009-5896(2015)02-0449-06
DOI: 10.11999/JEIT140512

Clustering Algorithms for Large-scale Social Networks Based on Structural Similarity

Chen Ji-meng[①] Chen Jia-jun[②] Liu Jie[①] Huang Ya-lou[②] Wang Yuan[①] Feng Xia[③]
[①](*College of Computer and Control Engineering*, *Nankai University*, *Tianjin* 300071, *China*)
[②](*College of Software*, *Nankai University*, *Tianjin* 300071, *China*)
[③](*Information Technology Research Base of CAAC*, *Civil Aviation University of China*, *Tianjin* 300300, *China*)

Abstract: To cluster the directed and large-scale social networks, a Structural Clustering Algorithm for Directed Networks (DirSCAN) and a corresponding Parallel algorithm (PDirSCAN) are proposed. Considering oriented behavioral relation between two vertices, DirSCAN is constructed based on action structural similarity and function analysis. To meet the need of large-scale social network analysis, a lossless PDirSCAN based on MapReduce distributed parallel architecture is designed to improve the processing performance. A large number of experimental results on real-world network datasets show that DirSCAN improves performance of SCAN up to 2.34% on F1, PDirSCAN runs 1.67 times faster than DirSCAN.

Key words: Social networks; Directed network clustering; Parallel algorithm; MapReduce

1 引言

随着博客、微博等社交媒体的兴起，以用户为节点、以用户关系为边的社交网络迅猛增长。用户的兴趣、行为、功能等关系使社交网络中存在多个社区或簇。为了发现网络中隐藏的簇结构，传统的网络聚类方法主要基于链接的稠密度(link-density)，使得簇内节点距离较近，簇间节点距离较远，如经典的Newman快速算法[1]和Kernighan-Lin算法[2]。然而，以上算法忽略了社交网络有向交互性和节点具有不同功能。一方面，社交网络中的节点关系是有向的，如微博中的关注关系，不同方向表明了不同的兴趣信息。另一方面，社交网络中节点具有不同功能，如连接多个簇的枢纽节点具有跨簇传播功能；孤立的离群节点在噪音检测、流失客户检测等任务中有重要作用。这两个结构特点对社交网络的理解和功能发现有重要的意义。

当前的社交网络聚类方法除了传统基于链接稠密度的方法[1–3]外，还包括考虑节点功能特性、网络的有向性等社交特性的聚类方法。另外，面向大规模社交网络的并行聚类方法也是目前重要研究方向之一。

文献[4]在链接稠密度的基础上，同时考虑结构

2014-04-22收到，2014-08-27改回
国家自然科学基金(61105049, 61300166)，中国民航信息技术科研基地开放课题基金(CAAC-ITRB-201303, CAAC-ITRB-201204)，天津市科技计划项目(13ZCZDGX01098)和天津市自然科学基金(14JCQNJC00600)资助课题
*通信作者：刘杰 jliu@nankai.edu.cn

相似度，提出 SCAN 算法，并分析节点功能。然而，该算法仅针对无向网络聚类，未考虑社交网络的有向性。考虑社交网络中的关系存在有向性，文献[5]将有向边转换为无向边，再使用传统的无向网络聚类方法聚类，然而该无向化方法损失了社交网络的有向结构特性。文献[6]将有向网络聚类问题转化成对有向图进行加权切割的优化问题进行解决。但是文献[5,6]算法并未区分网络中的节点功能。因此，本文基于 SCAN 提出有向网络聚类算法(DirSCAN)。

近年来，大规模网络数据的快速增长促进了动态增量和分布式并行聚类算法的研究。文献[7]提出一种随机游走与动态增量相关节点结合的网络聚类算法挖掘社区。文献[8]在 MapReduce 系统上设计了大数据并行聚类算法，采用抽样来减小数据。文献[9]提出一种基于社交关系的模糊聚类算法，辅助数据分布式存储，提升数据访问效率。然而，此类方法存在信息丢失，无法得到与原算法一致的结果。本文前期工作提出了并行的SCAN算法PSCAN[10]，可得与原算法等价的结果，与文献[11]类似。

本文创新点在于，考虑上述两方面，在识别簇与节点功能的 SCAN(Structural Clustering Algorithm for Networks)[4]算法基础上，设计了基于结构相似度的有向网络聚类算法 DirSCAN (Structural Clustering Algorithm for Directed Networks)。此外，近几年社交网络发展迅猛，海量节点及复杂关系的分析对单机串行方法是一个巨大的挑战。针对这种用户数上亿、关系复杂的大规模社交网络，本文基于 MapReduce[12]分布式并行架构将 DirSCAN 并行化，提出 PDirSCAN(Parallel DirSCAN)，在聚类结果一致下提高运行速度。

2 有向网络聚类算法 DirSCAN

在社交网络中，由节点主动发起的关联与节点本身的兴趣、行为直接相关，而节点被动接收的关联则表明了其他节点对该节点的兴趣，而非直接表明节点本身特性。如微博中用户 A 关注其感兴趣的用户 B，而 B 未关注 A，则 A 的兴趣偏好由 B 直接体现，而 B 则无法用 A 直接描述。在这种情况下，节点的出边较之入边更能反映节点信息。因此本文重点考虑节点的出边，提出结构相似度假设：若两个节点所能到达的节点越相似，则两节点属于同一簇的可能性越大。

2.1 DirSCAN 算法的基本定义

给定一个有向网络 $G=\{V,E\}$，V 为节点集合，E 为连接节点的有向边集合。从节点 v 到节点 w 的有向边记为 $<v,w>$，其中 $v,w\in V$。节点 v 的结构定义为从 v 出发一步到达的节点集合及其本身，记为 $\Gamma(v)$。

$$\Gamma(v)=\{w\in V\mid <v,w>\in E\}\cup\{v\} \tag{1}$$

根据结构相似度假设，两节点的到达节点重合越多则越相似，因此，两点之间的结构相似度定义为

$$\sigma(v,w)=|\Gamma(v)\cap\Gamma(w)|\Big/\sqrt{|\Gamma(v)|\cdot|\Gamma(w)|} \tag{2}$$

在社交网络中，如果用户 A 与用户 B 共同关注了一群相同的人，那么可认为 A 与 B 兴趣相似，我们将网络中兴趣相似的节点定义为到达邻居，如式(3)所示。

$$N_\varepsilon(v)=\{w\in\Gamma(v)\mid\sigma(v,w)\geq\varepsilon,\varepsilon>0\} \tag{3}$$

其中，ε 是用于划分邻居与非邻居的相似度阈值。若 $\varepsilon=0$，则所有到达节点均为邻居节点。

当一个节点拥有较多的到达邻居节点，我们认为其足够活跃，将其定义为核节点，用于扩大簇。

定义 1 核节点。若节点 v 的到达邻居节点个数超过某一临界值，则 v 为核节点，定义为

$$C_{\varepsilon,\mu}(v)\Leftrightarrow|N_\varepsilon(v)|\geq\mu \tag{4}$$

其中，μ ($\mu>0$)是活跃节点的到达邻居临界参数，用于判定核节点。

扩大簇的过程如定义 2 所示。

定义 2 直接结构可达。若一个节点 w 是一个核节点 v 的到达邻居节点，则 w 也应该与 v 属于同一个簇。我们将这一过程定义为 v 直接结构可达 w，即核节点与其到达邻居节点应属于同一簇，如式(5)所示。

$$\mathrm{DR}_{\varepsilon,\mu}(v,w)\Leftrightarrow C_{\varepsilon,\mu}(v)\wedge w\in N_\varepsilon(v) \tag{5}$$

2.2 DirSCAN 算法流程

接下来，介绍 DirSCAN 算法是如何工作的，包括如何实现簇的搜索以及如何分析节点的功能，枢纽和离群。第 1 步，将所有节点初始化为未分簇点；第 2 步，遍历所有核节点，并寻找核节点的直接结构可达节点，将它们合并为一个簇，根据簇中的核节点重复第 2 步再次扩展簇，直到没有新节点加入；第 3 步，遍历所有的未分簇节点，根据与其相邻的簇的数目将其分为枢纽点或离群点，有多个相邻簇的是枢纽节点，至多只有 1 个相邻簇的即为离群节点。具体算法如表 1 所示。

需要注意的是，DirSCAN 的最终分类结果对节点处理顺序不敏感。DirSCAN 算法与 SCAN 算法的不同之处在于两方面。一方面，DirSCAN 的结构相似度考虑了节点的到达邻居，即节点的出边这一

表 1 有向网络聚类算法 DirSCAN

输入：给定有向社交网络 $G=\{V,E\}$，参数 ε,μ

输出：社交网络的簇编号 c，枢纽节点标签 h，与离群节点标签 o

$G=\{V,E,\{c,h,o\}\}$

(1) 对 $\forall v\in V$

(2) { **IF** ($C_{\varepsilon,\mu}(v)$)

(3) { 分配一个新的簇编号 c 给 v;

(4) 将 v 的到达邻居 $x\in N_{\varepsilon}(v)$ 插入队列 Q;

(5) **WHILE** 队列 Q 不为空 $Q\neq 0$ **DO**

(6) { y 取队列 Q 中第 1 个节点;

(7) $R=\{x\in V\mid \mathrm{DR}_{\varepsilon,\mu}(y,x)\}$;

//将 y 直接结构可达的节点集合加入同一簇中

(8) **FOR** $\forall x\in R$ **DO**

(9) { **IF** (x 未分类) //若节点 x 未处理，

//将其加入队列 Q 中

(10) 将 x 插入 Q;

(11) **IF**(x 为未分类或 NM)

(12) 将 x 分为 c;

}

(13) 将 y 从 Q 中移除;

}

}

(14) **ELSE** 将 v 标记为无簇节点 NM;

}

(15) 对 $\forall v\in\{\mathrm{NM}\}$

(16) { **IF** ($(\exists x,y\in \Gamma(v))\wedge(x.c\neq y.c)$)

(17) 将 v 记做枢纽节点 h;

(18) **ELSE** 将 v 记做离群节点 o;

}

有向传播特性；另一方面，由于 DirSCAN 采用有向边来定义直接结构可达性，因此该特性不可逆。这两方面的考虑使得本文所计算的结构相似度更能反映真实社交网络的情况。

2.3 DirSCAN 算法的复杂度分析

DirSCAN 算法仅需遍历有限次节点和边，一次遍历即可获得节点的到达邻居、判断核节点，从而以核节点进行簇扩展。因此若网络中存在 n 个节点，遍历节点的复杂度为 $O(n)$。在遍历边时，需要计算节点的每条出边是否为到达邻居关系，最差情况为所有节点都相连，复杂度为 $O(n(n-1))$。由于实际社交网络通常为稀疏网络[4]，遍历边的次数可近似为遍历节点的次数。因此 DirSCAN 算法的时间复杂度近似为 $O(n)$。

3 并行有向网络聚类算法 PDirSCAN

为了适应大规模社交网络的聚类，本节将在 MapReduce 并行平台上设计并行化算法 PDirSCAN。

通过分析发现，DirSCAN 算法对节点的操作主要分为两个步骤：识别到达邻居与核节点；扩充簇以完成聚类。第 1 步中，每个节点都可以独立计算到达邻居和节点间的结构相似度。第 2 步中，每个核节点可独立将其标签传递给其到达邻居。可见，DirSCAN 算法并行化是可行的。

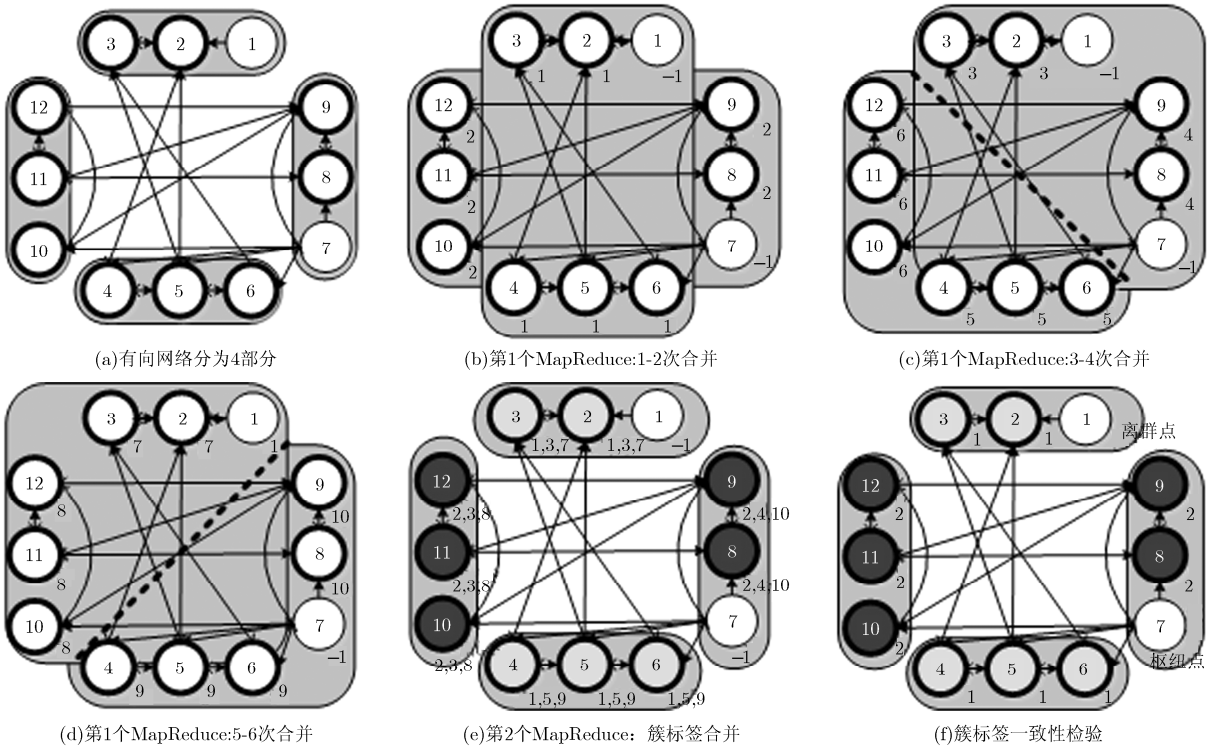
MapReduce 的并行数据处理过程可分为两个步骤：Map 和 Reduce。Map 将输入的<key, value>对映射到新的<key, value>对上，用来将数据打散成多组子数据。Reduce 独立并行地处理各组子数据。MapReduce 自身提供了很好的容错性，使得整个任务不会因为某个处理节点的瘫痪而整体崩溃。

3.1 PDirSCAN 中识别到达邻居的并行化

并行识别节点到达邻居这一步骤由两个 MapReduce 任务来实现。第 1 个 MapReduce 任务并行计算每个节点与其临近点之间的到达邻居关系，如图 1(a)~1(d)所示。其中 Map 函数将网络随机切分成若干份，然后复制多个副本，将其两两合并形成对，假设网络被分割成 4 份，则需要 6 次合并。Reduce 函数在本地计算每个节点的到达邻居。第 2 个 MapReduce 任务对每个节点的所有到达邻居进行汇总，仅进行 Reduce 步骤，如图 1(e)所示。其中 Reduce 函数将所有中间数据进行排序，排序后可依次将含同一节点的数据聚合。

3.2 PDirSCAN 中簇扩展的并行化

当获得了所有节点的到达邻居之后，可判断该节点是否为核节点，随后进行簇扩展。在这一过程中，通过核节点来传播簇标签以获得最终的结果，可由两个 MapReduce 任务完成。第 1 个任务将数据随机划分为若干份(如图 1(a)所示，其中粗边节点是核节点)，将多个副本进行两两合并，扩展簇标签(如图 1(b)~1(d)所示，节点右下角为节点所属的簇标签，其中“-1”为处理过但未分配簇的节点)。第 2 个任务将所有聚类后的簇标签合并，实现标签的全局传播及聚类，如图 1(e)~1(f)所示。由于相同簇节点在不同机器上聚类的簇标签不一致，如图 1(e)所示，同簇中的节点曾被聚为 2, 4, 6, 8, 10 簇，因此本文将簇标签索引列表中的最小标签作为该簇的标签完成标签一致化，如图 1(f)，其中簇标签索引列表记录相同簇中所有节点曾标记过的簇标签。最后，获得最终的簇集合。那些无簇标签的节点则根据其到达邻居的簇类别数标记为枢纽点或离群点。



(a)有向网络分为4部分 (b)第1个MapReduce:1-2次合并 (c)第1个MapReduce:3-4次合并

(d)第1个MapReduce:5-6次合并 (e)第2个MapReduce：簇标签合并 (f)簇标签一致性检验

图 1 聚类并行过程细节

3.3 PDirSCAN 的算法复杂度分析

假设有向网络中，有 n 个节点，被 p 台机器划分成 m 份。由 DirSCAN 的算法复杂度可知串行计算的时间复杂度为 $T_s = O(n)$，则并行后数据处理时间复杂度为 $O(n/p)$。假设通信之前的同步用时为 T_0，由于每个节点都需要至少传送到其他节点一次，因此并行时通信用时为 $T_c = T_0 + O(n(m-1)/2)$。综上所述，PDirSCAN 总复杂度为，$T_p = T_0 + O(n(m-1)/2) + O(n/p)$。若通信用时 T_c 小于串行计算用时 T_s，则并行计算时间复杂度优于串行计算。由于社交网络大都是稀疏网络，因此通信用时较少，并行算法存在速度优势。

4 实验与分析

4.1 实验数据集

本文在两个真实网络数据集上进行实验。在网络数据集 WebKB[13]上，进行有向网络聚类的准确性实验，对比分析 DirSCAN 与 SCAN。在大规模的社交网络数据集 Pokec[14]上，进行 PDirSCAN 的并行效率实验。

WebKB 数据集包含了 Texas, Washington, Cornell, Wisconsin 这 4 所大学网页之间的链接情况，包含 877 个节点和 1608 个有向边。这些网页可分为 5 类：课程，教师，员工，学生以及项目。

Pokec 大规模社交网站数据集记录了斯洛伐克的好友关注关系网络，包含 1632803 个节点和 30622564 条有向边，平均节点出度为 18.8。Pokec 没有真实分类，因此仅用于测试并行实验中的效率。

4.2 评价指标介绍

准确性实验采用聚类常用的评价指标准确率(Precision, P)、召回率(Recall, R)、F1 值和边缘索引 (Rand Index, RI)来评价聚类结果的准确程度。真实情况下将同类两个节点聚为一簇，为一个正确的聚类结果。这 3 个评价指标的值越大表明聚类结果与真实情况越相似，聚类效果越好。

在并行效率实验中，我们采用并行实验中的常用评价指标加速比(speedup)、规模增长性(sizeup)和可扩展性(scaleup)进行度量。加速比指串行与并行处理最短用时之比，加速比越大说明并行用时越短。规模增长性是指并行计算 m 倍数据量与单倍数据量的时间比，该指标越小说明数据增多用时增长慢。可扩展性是指在单机上处理单倍数据量与在 m 台机器上处理 m 倍数据量的时间比，该指标越大表明可扩展性越好。

4.3 实验设置

本文采用 SCAN 作为对比算法。SCAN 只适用于无向网络聚类，因此先将有向网络转换为无向网络。算法中的参数 ε 将遍历[0,1]中步长为 0.1 的数值来进行优化，μ 将遍历[1,10]中步长为 1 的数值来进行优化。

4.4 实验结果及分析

4.4.1 DirSCAN 聚类算法的准确性实验结果　在 WebKB, Texas, Washington 数据集上的聚类准确率实验结果如表 2 所示。结果显示，考虑了网络有向性的 DirSCAN 算法，在准确率 P、召回率 R、F1 值和 RI 上都优于无向图聚类算法 SCAN，分别提高 0.39%, 8.83%, 2.34%和 0.88%。其中，召回率 R 和 F1 值提升最明显。在 WebKB 各大学的子数据集上也有相似结果，Texas 子数据集中 DirSCAN 在召回率 R、F1 值上分别提升 16.98%, 7.05%, Washington 子数据集中 DirSCAN 在召回率 R、F1 值上分别提升 11.44%, 3.05%，可见，考虑网络有向性对聚类有效。

4.4.2 PDirSCAN 并行化算法的效率实验结果　为了验证 PDirSCAN 的并行效率，本文在 4 台计算机上进行实验。每一台机器的处理器都为 2.59 GHz AMD Phenom(tm) II X4 810, 3G 内存。本文将 Reduce 任务的数目设置成与集群的机器数目相同，即每一台机器处理至多一个 Reduce 任务。在所有并行实验中，数据集都被分成 24 份，保证所需要合并的次数相同。多次实验验证，并行实验结果与串行一致[11]。

在 Pokec 数据集上的并行效率实验结果如图 2 所示。实验表明，(1)当节点数量不变时，加速比随机器数目增多而提高，说明所需的处理时间减少了；当节点数增加时，加速比增加更显著，在 8×10^5 节点 4 台机器，比单机处理速度提高了 1.67 倍，见图 2(a)。(2)单机处理节点时，规模增长性提升较快，即节点增加使处理时间增长(8×10^5 节点比 1×10^5 节点耗时多了 1.87 倍)，而当机器数增加时，规模增长性提升缓慢，即时间消耗无显著增加，使用 4 台机器时， 8×10^5 节点比 1×10^5 节点耗时仅多了 1.28 倍，比单机快了 0.59 倍，见图 2(b)。(3)当机器数目与数据量等比增加时，可扩展性提高至 1.1，即若单机处理 1×10^5 节点需费 t 时，4 台机器采用 PDirSCAN 可仅用 $0.9t$ 的时间处理 4×10^5 节点，见图 2(c)。

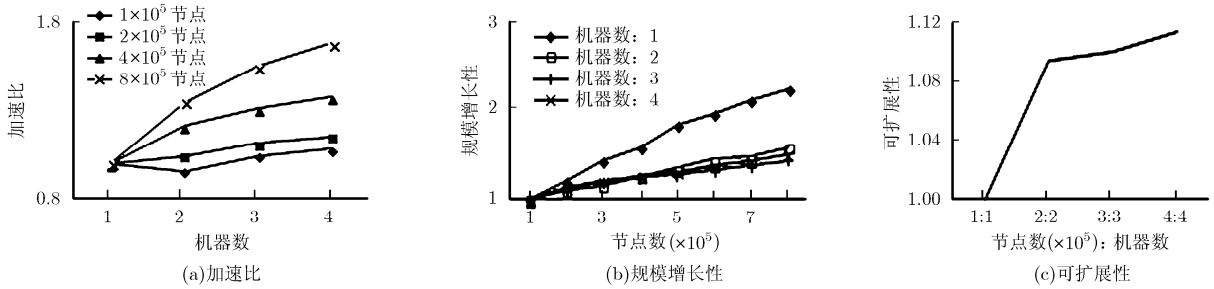
综上所述，PDirSCAN 在聚类结果与 DirSCAN 一致下，提高了处理速度，有较高的实际应用价值。

5　结束语

本文针对社交网络的有向交互性，提出基于结构相似度的有向网络聚类方法 DirSCAN 来检测社区，F1 值可提升 2.34%。针对真实网络大规模特性，本文提出基于 MapReduce 的并行化算法 PDirSCAN 提高算法速度 1.67 倍。实验结果表明本文算法提高了网络聚类的效率和速度，具有较大的实用价值。

表 2 两种算法在 WebKB, Texas, Washington 数据集上的聚类结果(%)

算法	WebKB				Texas				Washington			
	P	R	F1	RI	P	R	F1	RI	P	R	F1	RI
SCAN	30.59	56.57	39.71	45.91	30.54	31.25	30.89	48.29	30.93	48.32	37.72	49.02
DirSCAN	30.98	65.41	42.05	46.78	31.28	48.23	37.95	49.25	30.93	59.76	40.76	48.16



(a)加速比　(b)规模增长性　(c)可扩展性

图 2　PDirSCAN 在 Pokec 数据集上的并行结果

参 考 文 献

[1] Newman M E J. Fast algorithm for detecting community structure in networks[J]. *Physical Review E*, 2004, 69(6): 066133-1–066133-5.

[2] Lancichinetti A, Fortunato S, and Kertész J. Detecting the overlapping and hierarchical community structure in complex networks[J]. *New Journal of Physics*, 2009, 11(3): 033015-1–033015-18.

[3] Fallani F D V, Nicosia V, Latora V, *et al*.. Nonparametric resampling of random walks for spectral network clustering[J].

Physical Review E, 2014, 89(1): 012802-1–012802-5.

[4] Xu Xiao-wei, Yuruk N, Feng Zhi-dan, *et al*.. SCAN: a structural clustering algorithm for networks[C]. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, 2007: 824–833.

[5] Zhou Deng-yong, Huang Jia-yuan, and Schölkopf B. Learning from labeled and unlabeled data on a directed graph[C]. Proceedings of the 22nd International Conference on Machine Learning, Bonn, 2005: 1036–1043.

[6] Meila M and Pentney W. Clustering by weighted cuts in directed graphs[C]. Proceedings of the 7th SIAM International Conference on Data Mining, Minneapolis, 2007: 135–144.

[7] 肖杰斌, 张绍武. 基于随机游走和增量相关节点的动态网络社团挖掘算法[J]. 电子与信息学报, 2013, 35(4): 977–981.
Xiao Jie-bin and Zhang Shao-wu. An algorithm of integrating random walk and increment correlative vertexes for mining community of dynamic networks[J]. *Journal of Electronics & Information Technology*, 2013, 35(4): 977–981.

[8] Ene A, Im S, and Moseley B. Fast clustering using MapReduce[C]. Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, 2011: 681–689.

[9] Cao Yan, Cao Jian, and Li Ming-lu. Distributed data distribution mechanism in social network based on fuzzy clustering[J]. *Foundations and Applications of Intelligent Systems*, 2014, 213: 603–620.

[10] Chen Jia-jun, Chen Ji-meng, Liu Jie, *et al*.. PSCAN: a parallel structural clustering algorithm for networks[C]. Proceedings of the 2013 International Conference on Machine Learning and Cybernetics, Tianjin, 2013: 839–844.

[11] Zhao Wei-zhong, Venkataswamy M, and Xu Xiao-wei. PSCAN: a parallel structural clustering algorithm for big networks in mapReduce[C]. Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications, Washington DC, 2013: 862–869.

[12] Dean J and Ghemawat S. MapReduce: simplified data processing on large clusters[J]. *Communications of the ACM*, 2008, 51(1): 107–113.

[13] Craven M, DiPasquo D, Freitag D, *et al*.. Learning to extract symbolic knowledge from the world wide web[C]. Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98), Madison, 1998: 509–516.

[14] Takac L and Zabovsky M. Data analysis in public social networks[C]. Proceedings of International Scientific Conference & International Workshop Present Day Trends of Innovations, Lomza, 2012: 1–6.

陈季梦： 女，1987 年生，博士生，研究方向为数据挖掘.
陈佳俊： 男，1988 年生，硕士，研究方向为并行与分布式计算.
刘 杰： 男，1979 年生，博士，副教授，研究方向为机器学习.