基于启发式函数的分布式 FN 算法①

肖有诰, 屠成宇

(江南计算技术研究所, 无锡 214083)

摘 要: 对复杂网络进行社团挖掘和分析是很多领域和学科的重要问题, 结合海量数据通联矩阵稀疏的特点, 提出了一种基于启发式函数合并的快速社团挖掘算法 KFN 算法, 并建立了算法的 MapReduce 模型. 通过对 DBLP 论文合著网络数据集进行挖掘可知, 分布式模型和基于启发式函数的合并策略能够提高社团挖掘的时间效率.

关键词: 海量数据; 社团挖掘; 启发式函数; MapReduce

Distributed FN Algorithm Based on Heuristic Function

XIAO You-Gao, TU Cheng-Yu

(Jiangnan Institute of Computing Technology, Wuxi 214083, China)

Abstract: The mining and analysis of community in complex networks is an important issue in many domains and disciplines. In this paper, focus on the sparsy Communication Matrix of the massive data, we suggest a fast mining algorithms based on a heuristic function to merge called KFN algorithm, and also show out the MapReduce model of this algorithm. With experiment focued on the DBLP paper co-network data sets, we conclude that distributed mining model and the merging strategy based on the heuristic function can improve the time efficiency on community mining.

Key words: massive data; community mining; heuristic function; MapReduce

社团挖掘(Community Mining, CM)由于其巨大的应用价值成为近年来数据挖掘热门的课题之一, 是根据数学方法、图论等发展起来的分析方法[1], 旨在发现社会网络中在某些方面具有相似特点的实体组成的相对独立和封闭的团体. 社会网络是指由个体及个体之间的关系构成的满足社会结构特点的网络, 社会中人与人交往关系构成的网络, 网页链接关系构成的网络, 电信通话网络, 文献引用关系构成的网络等都是典型的社会网络. 互联网和现代通信技术的发展, 使得社会网络容量越来越大, 社团数据呈现海量的特点. 面对海量数据挖掘, 计算性能不能满足挖掘的实时性问题比较突出, 本文拟从分布式计算[2]和对社团挖掘算法 FN 进行改进, 提出了 FN 算法的 MapReduce 模型和基于启发式函数合并策略的 KFN 算法, 然后对 DBLP 论文合著网络数据集进行社团挖掘实验.

1 系统概述

社团指网络中节点的集合, 社团中的节点之间具有紧密的连接, 而社团之间则为松散的连接. 图 1 为一个社团的示意图, 图中有两个社团, 社团内部节点联系紧密而社团之间的联系松散.

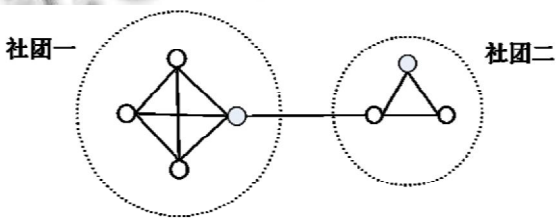

图 1 社团结构示意图

社团网络以图的方式来建模, 它可以抽象为一张具有 n 个节点和条边的图, 常用 $G=(V,E)$ 来表示, V 和

① 收稿时间:2012-02-28;收到修改稿时间:2012-04-11

E 分别是节点和边的集合. 它可以用一个 $N \times N$ 的矩阵来表示.

如果 $A=(A_{ij})n \times n$ 是网络的邻接矩阵, 那么

$$A_{ij}=\begin{cases}1 & \text{如果节点i和j是相连的}\\ 0 & \text{其他情况}\end{cases}$$

2 社团挖掘算法

2.1 FN 算法

FN 算法是 Newman 于 GN[3]算法之后提出的一种快速挖掘算法. 为了得到具有实际意义的社团结构, Newman 等人定义了社团模块度函数 Q 来衡量网络划分质量, Q 函数的定义为: 网络社区内实际存在的边数与完全随机的连接情况下社区内期望的边数之差. 给定一个无向无权网络 $N=(V,E)$, 假设点集 V 被划分为若干个社团. 若网络中任一结点 i 所属的社区为 $r(i)$, 则 Q 函数[4]可被定义为,

$$Q=\frac{1}{2m}\sum_{ij}((A_{ij}-\frac{k_i k_j}{2m})\times\delta(r(i),r(j)))$$

其中, $A=A(i,j)_{n*n}$ 表示网络 N 的邻接矩阵, 如果结点 i 与 j 之间存在边连接, 则 $A_{ij}=1$, 否则 $A_{ij}=0$. 可见, Q 值越大, 社团内部成员成为一个社团可能性越大; 反之亦然.

快速算法以这样一种状态开始, 即每个节点是 N 个社团的单独的成员, 选择合并使 Q 值增大最多或减少最小的社团对进行合并. 现有 A、B 两个社团合并, 如图 2 所示, a_i 为社团 A 中的成员, b_j 为社团 A 中的成员, 合并后的 Q 的改变量用 ΔQ 表示, ΔQ 为一个矩阵, 其元素 e_{ij} 为社团 i 和社团 j 合并后 Q 的变化值,则

$$\Delta Q_{AB}=Q_{AB}-Q_A-Q_B=\frac{1}{2m}\sum_{a_i\in A,b_j\in B}((A_{ij}-\frac{k_i k_j}{2m})\times\delta(r(i),r(j)))$$

社团合并后, 因为合并无边相邻的社团对不能导致 Q 的增加, 只需要考虑那些相互之间有边的社团对.

在合并之后, 对应的 ΔQ 矩阵元素 e_{ij} 需要更新. 若选择让 A、B 社团合并为社团 D, 则新社团 D 与其他社团之间(如社团 C)的合并关系如图 2 所示. 在计算新一轮合并 Q 时, 易知 $\Delta Q_{DC}=\Delta Q_{AC}+\Delta Q_{BC}$. 上一次的迭代计算可以被下一轮的计算所采用, 这将是算法实现中很重要的一点, 只需要将 Q 矩阵中与 i, j 社团相关的行和列相加, 时间复杂度为 $O(n)$, 如此, 算法的每一步的时间复杂度为 $O(m+n)$. 最多要执行 $n-1$ 次合并来构成完全的系统树图, 因此整个的算法运行时间是 $O((m+n)*n)$, 或在稀疏图中是 $O(n^2)$.

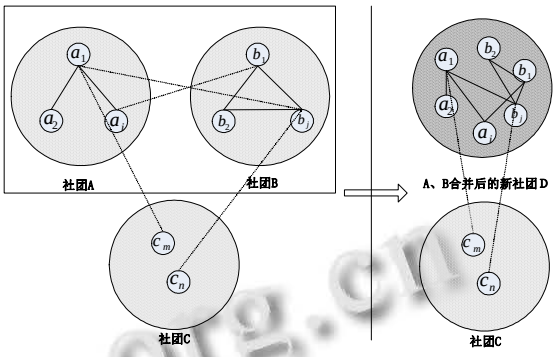

图 2 社团合并图示

算法执行过程如下[5]:

(1)初始化网络为 n 个非空社团(图中共 n 个节点), 初始的 e_{ij} 和 q_i 如下:

$$e_{ij}=\begin{cases}1/2m, \text{如果节点 } i\text{和}j\text{之间有边相连}\\ 0, \text{其它}\end{cases}$$

$q_i=k_i/2m$, 其中 k_i 为节点 i 的度, m 为网络的总边数.

(2) 第一次合并有边相连的社团对, 并计算 $\Delta Q_{ij}=2(e_{ij}-q_i q_j)$;

(3)选择中 ΔQ 最大值对应的两个社团进行合并, 并更新 ΔQ 矩阵;

(4)迭代执行步骤 3, 不断合并社团, 直到没有相关联社团为止, 最多要执行 n-1 次合并.

2.2 基于启发式函数合并的 KFN 算法

在实际应用中, 社团关系比较稀疏, 且经验说明, 最后的独立的社团比较多, 仅仅每次只选择最大的一对社团进行合并显得不太合理. 加快迭代的速度是加快算法收敛的最直接手段, 每次合并 K 对社团, 相当于将迭代的次数进行了 K 次, 同时在海量数据组成的海量社团应用中, 社团的模块化往往一开始体现的最为明显, 因为初始的联系密度会比较大. K 对合并的 FN 算法理论上是可行的, 在课题实际应用中, 提出了基于启发式函数合并的 KFN 算法.

假设第 n 次合并时候, 共选择了 K_n 对社团进行合并, 此次合并的 K_n 个社团对对应的模块化增量向量表示为 $(q_1,q_2,...,q_{K_n})$, $q_{\max N}, q_{\min N}, q_{aveN}$ 分别为此向量的最大值、最小值和平均值. 则第 n+1 次合并时候, 合并对

数的启发式函数为:

$$K_{n+1}=\begin{cases}K_n*(1+\dfrac{q_{\max(N+1)}-q_{\min N}}{\left|q_{\max(N+1)}-q_{aveN}\right|}),\text{若}K_n*(1+\dfrac{q_{\max(N+1)}-q_{\min N}}{\left|q_{\max(N+1)}-q_{aveN}\right|})\geq 1\text{且n}>0;\\ K_0,\text{若}K_n*(1+\dfrac{q_{\max(N+1)}-q_{\min N}}{\left|q_{\max(N+1)}-q_{aveN}\right|})<1\text{或n}=0\end{cases}$$

其中 K_0 为初始值. 启发式函数根据相邻两次的模块化增量向量 $(q_1,q_2,...,q_{K_n})$ 来决定每次合并的社团对个数.

KFN 算法同时体现了社团“团内聚, 团间松”的特征, K 对合并策略使我们将算法时间偏向社团内部节点的挖掘, 而并不是将注意力集中在边缘点上. 通过对边缘节点的合理忽视, 可以节约更新时间. KFN 算法描述如下:

(1)初始化网络为 n 个非空社团(图中共 n 个节点), 初始的 e_{ij} 和 q_i 如下:

$$e_{ij}=\begin{cases}1/2m,\text{如果节点 }i\text{和}j\text{之间有边相连}\\ 0,\text{ 其它}\end{cases}$$

$q_i=k_i/2m$, 其中 k_i 为节点 i 的度, m 为网络的总边数. e_{ij} 是社团中边的比率, 第一步的初始值可随意设定;

(2) 第一次合并有边相连的社团对, 并计算 $\Delta Q_{ij}=2(e_{ij}-q_iq_j)$ 和 K_n;

(3)选择前 K_n 个 ΔQ 对应的两个社团对合并, 并更新 ΔQ 矩阵;

(3)迭代的执行步骤 3, 不断合并社团, 直到没有相关联社团为止.

如启发式函数所示, 算法最坏每步只合并一对社团, 时间复杂度为 $O((m+n)*n)$, 或在稀疏图中是 $O(n^2)$; 理想情况下算法每次合并对数大于 K_0, 时间复杂度小于 $O((m+n)*n/K_0)$ 或在稀疏图中小于 $O(n^2/K_0)$.

3 KFN算法的MapReduce模型

MapReduce[6]是 Google 公司的核心计算模型, 它将复杂的运行于大规模集群上的并行计算过程高度地抽象到两个函数 Map 和 Reduce. 用户只需要提供自己的 Map 函数以及 Reduce 函数就可以在集群上进行大规模的分布式数据处理. 适合用 MapReduce 来处理的数据集有一个基本要求: 待处理的数据集可以分解成许多小的数据集, 而且每一个小数据集都可以完全并行地进行处理.

社团挖掘算法是一种图合并算法, 在图算法中, 由于节点之间的交互比较多, 计算复杂度往往都很高. 同时, 一般需要将图的关联关系缓冲到一个队列中, 所以图算法也是一个耗内存的过程. 在一个 8G 内存的服务器上, 当通联记录 800 万条时候, 就出现了内存被耗光以致程序错误的情况. 分布式图算法, 通过将大图分割成小图, 分别对小数据快进行计算进而合并, 以准确性为代价可以有效提高算法的时间效率. 在图的分布式计算中, 小粒度的分割会提高结果的准确性但是计算复杂度很高, 大粒度的分割会降低计算复杂度但往往会获得较低的准确性. 同时, 在内存一定的情况下, 可以通过增加服务器的数量满足通联数据量不断增长的需求.

3.1 数据未分割的 KFN 算法

图 3 以流程图的形式展现了算法合并后更新 deltaQ 矩阵的 MapReduce 过程:

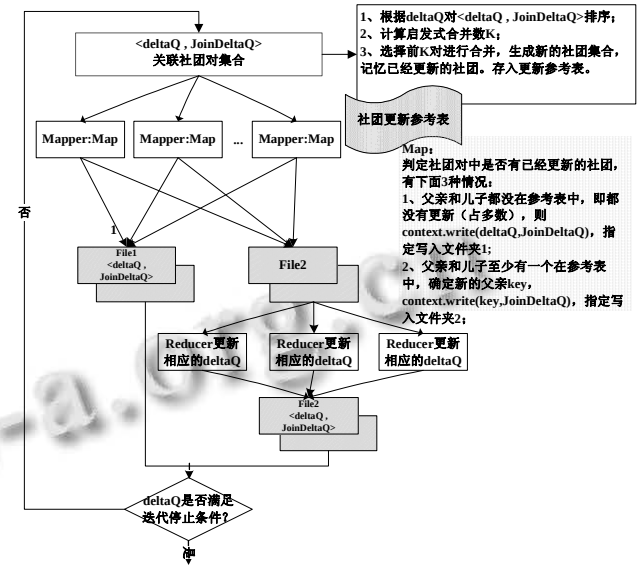

图 3　更新 deltaQ 矩阵的 MapReduce 处理流程图

3.2 数据分割的 KFN 算法

如果单台机器的最大数据处理能力不能满足挖掘需求, 可以通过拓展计算节点的方法增加其处理能力. 数据分割的 KFN 算法先将初始的通联数据进行分块, 将各块数据发布到一个独立的 Map 节点中, 多个 Map 任务并行运行 KFN 算法进行社团挖掘, 最后再对结果进行 Reduce 操作, 图 4 为数据分割的 KFN 算法的分布式计算流程图:

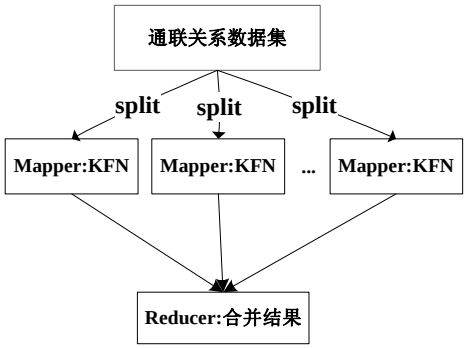

图 4 分割 KFN 算法的分布式计算流程图

3.3 实验结果分析

分别在单机和分布式环境下对 DBLP 网络数据集数据进行挖掘实验, 分布式环境是由 1 个管理节点, 2 个计算节点构成. 服务器的配置都为: CPU 频率(GHZ): 1.8G; 内存容量: 8G; 硬盘: 250GB; 操作系统: Linux. DBLP 网络是从 2007 年 DBLP 计算机科学在线论文网络中抽取的包含数据库、信息检索、数据挖掘和机器学习这 4 个研究方向学者合著信息的一个真实网络.

DBLP 官网上提供了 1.3G 的 XML 格式数据 dblp.xml. 采用 sax 对其解析, 经过去重后, 得到 120170 个节点和 155572 条边. 在单机上用 FN 算法进行挖掘, 耗时 159s, FN 算法的最后运行结果为 7289 个社团. 设定整个社团的模块化函数 $\sum_i Q_{\bar{i}} > 0.5$ 为合并的截止条件, 根据先验知识, 对机器学习领域 Michael I. Jordan、信息检索领域的 James P. Callan、数据库领域的 Jeffrey D. Ullman 和数据挖掘领域 Charu C. Aggarwal 等四人[7]所在的社团进行逆向查找, 得到如下结果:

表 1 Q=0.499 时社团信息

研究领域	查找人	社团成员数	社团 Q 值
机器学习	Jordan	4166	0.163
数据挖掘	Aggarwal		
信息检索	Callan	2256	0.087
数据库	Ullman	2945	0.126

由表 1 可知, 用 FN 算法挖出的社团能体现出研究领域的特点, 从社团的 Q 值可分析出这三个社团也占据全部网络数据集凝聚度函数 $\sum_i Q_{\bar{i}}$ 的主体部分, 由于某个中间合作者的原因, Jordan 和 Aggarwal 共处一个社团, 这与 FN 算法并不矛盾, 因为 FN 算法提供的是一个合并顺序的度量方法.

当 K 的初始值依次取 1、2、4、6、8, 用 KFN 算法进行社团挖掘, 时间测试结果如下图 5 所示:

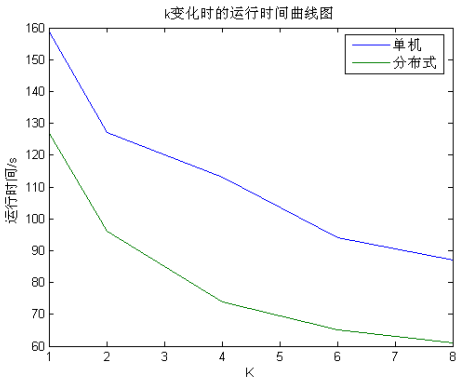

图 5 KFN 算法运行时间曲线

由上图可知, 分布式模型可以提高社团挖掘的时间效率, 随着 K 值得增加, 算法的收敛速度会加快, 基于启发式函数的 KFN 模型和分布式算法都可以提高社团挖掘的时间效率.

4 结语

本文针对海量通联数据的社团挖掘需求, 提出了分布式社团挖掘方法. 并且结合海量数据的通联矩阵稀疏性等特点, 提出了基于启发式函数合并的 KFN 算法. 实验证明, 分布式算法和 KFN 算法都是有效的. 但是数据分割的 MapReduce 模型是以牺牲结果准确性为代价, 进而提高时间效率的. 如何同时提高分布式 KFN 算法时间性能和准确性能, 还需要不断的探索分析.

参考文献

1 吴文涛.图对称理论在社会网络分析若干重要问题中的应用.复旦大学,2010.
2 信息架构本质:分布式数据挖掘. http://www.ibm.com/developerworks/cn/architecture/ar-infoarch6/
3 Girvan M, Newman EJ. Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America; 6/11/2002, Vol. 99 Issue 12, p7821, 6p, 7 Diagrams, 1 Graph
4 朱小虎.用于社团发现的 Girvan-Newman 改进算法.计算机科学与探索,2010,4(12):1101−1108.
5 行花妮,刘刚,王磊.基于 GN 算法的快速算法在 PPI 网络中的实现.计算机与信息技术,2009.9.
6 Amazon Elastic Map Reduce. [2010-09-26].http: //aws. Amazon.com/ elasticMapReduce/.
7 黄健斌,孙鹤立, Bortner D,刘亚光.从链接密度遍历序列中挖掘网络社团的层次结构.软件学报,2011,22(5):951−961.