

一种从马尔可夫聚类簇发现潜在 WEB 社区特征的方法

杨楠 林松祥 高强 孟小峰

(中国人民大学信息学院 北京 100872)

摘要 在分析了目前一些典型的社区发现算法的基础上,通过对无主题条件下的隐含社区发现算法的研究,提出将基于流的社区特征和马尔可夫图聚类算法(MCL)的簇结合起来寻找 Web 隐含社区的方法.将镜像或近似镜像页面的删除放在图聚类之后,大大减少了比较的代价.然后,在聚类簇的基础上,使用判定每个簇内元素的筛选算法产生可能的社区候选集合.实验表明,该方法是可行的,可以发现许多存在的社区.

关键词 Web 社区;链接分析技术;MCL 图聚类;流量模拟;随机漫游
中图法分类号 TP393

Discovering Signature of Potential Web Communities from Clusters of MCL

YANG Nan LIN Song-Xiang GAO Qiang MENG Xiao-Feng

(School of Information, Renmin University of China, Beijing 100872)

Abstract Web community is an important social activity in the evolution of Web. The paper analyzes typical algorithms of present Web communities' discovery. Under the condition of non-topical pre-defined and implicit communities, a new method is proposed, which combine both characteristic structure of community and the clusters of Markov Graph Clustering(MCL) to find implicit communities. The procedure of deleting mirror or near-mirror pages is arranged behind graph clustering so that decrease comparing cost considerably. Then a community member select algorithm is used to produce the set of community candidates. The experimental results show the new method works properly and many Web communities are inferred.

Keywords Web community; link analysis; MCL graph clustering; flow simulation; random walk

1 引言

Web 社区是 Web 发展过程中一个重要的社会活动现象,体现为大量的页面被众多的超级链接紧密连接在一起.通常这些聚集在一起的页面集合隐含着一个或者多个主题.Web 社区的形成是一些具有相同兴趣的个人或者组织建立的页面集合,在这些页面的内部通常会嵌入一些表明建立者观点的超级链接,指向他们认为是价值的网页.因此,发现

这样的 Web 社区,对我们了解 Web 中的社会活动、Web 的信息构成和结构化的组织方式、在 Web 中的信息搜索、合理地组织 Web 信息结构、商家分析客户的构成、网站门户的有效组织等提供了良好的基础.

Web 社区有助于人们对 Web 的认识,并且有利于引导人们对信息的搜寻(例如,表现为新闻组(newsgroup)、网圈(Webbring),或者如 Yahoo!和 Infoseek 中目录形式的资源集合以及如 Geocity 中的社民等等^[1]),使得网络用户可以直接进入自己感

收稿日期:2005-09-08;修改稿收到日期:2007-01-26. 本课题得到教育部 211 项目子课题《WEB 资源发现技术》的资助. 杨楠,男,1962 年生,副教授,研究方向为 Web 数据挖掘、Web 知识发现、图聚类和信息检索. E-mail: yangnan@ruc.edu.cn. 林松祥,男,1981 年生,硕士研究生,研究方向为 Web 社区发现技术、图聚类. 高强,男,1981 年生,硕士研究生,研究方向为 Web 社区发现技术、Web 爬取技术. 孟小峰,男,1964 年生,博士,教授,博士生导师,主要研究领域为数据库系统、Web 数据集成、XML 数据管理、移动数据库等.

兴趣的社区,并且在社区的内部进行搜索,大大减少了一些不必要的返回结果,提高了查询的效率。

目前,Web 中存在着大量的明确定义的、知名的社区,如我们在上面提到的。这些明确定义的社区基本上是靠人为发现和人工维护的,如 Yahoo!和 Infoseek 等请一些分类专家对 Web 信息进行分类,分层组织成为树形结构,而用户则喜欢通过结构树的方式浏览页面,因为这些树结构是由人维护的较高质量的信息索引树。

虽然人工维护的结构树对于许多主题搜索非常有效,但是该结构树一般都是主观建立的。结构树的建立和维护极为昂贵,不但改进缓慢,而且无法覆盖窄范围的主题。另外,由于这些社区的数量巨大,同时 Web 社区一般并没有在互联网上明确定义,并且许多社区处于初期形成阶段,许多现有的社区也在不断更新。因此,通过人工的方式去跟踪这些社区是非常困难的。由于这些潜在的隐含社区的数量大大超过明确定义社区,因此,人工明确定义的社区无法识别出这些潜在的社区,而且潜在社区的数量还在不断地增加。而许多的情况是,潜在社区往往是在某个参与者还没有意识到它存在之前就形成了。

因此,需要开展自动化的或半自动化的社区发现技术和社区的维护技术的研究。发现技术主要针对那些潜在的、即将形成的社区,而维护技术是对现有的社区结构树的更新技术。随着社区自动发现技术研究的不断开展,已经出现了一些网上社区的发现技术^[1-7]。本文通过对现有的社区发现技术的分析,提出了将图形聚类的算法应用到社区发现技术当中的方法,它可以高效和快捷地发现 Web 中的社区。

本文第 2 节介绍相关的研究工作;第 3 节介绍基本的 Web 图知识和图形聚类 MCL 方法;第 4 节描述 MCL 方法应用在社区发现的实现细节;第 5 节介绍实验的组织过程和结果的分析;第 6 节是结论和未来的工作。

2 相关研究

由于社区的发现对于 Web 信息检索有很大的帮助,所以,许多研究都在致力于社区的发现技术。总的来说,这些技术基本上都是以 Web 链接分析为基础的,都采用了 Web 的结构对社区的影响来发现社区的特征,基本上都是通过将 Web 看成一个巨大的图来处理的。

按照 Kleinberg 的观点,社区特征的定义大致为两类:一类为通过相互连接的 hub 和 authority 集合,来发现社区的链接结构的特征(具有二分特征)。另一类是将社区看成是一些紧密连接页面的集合,也就是社区内部节点之间的链接数量要大于到达社区外的链接数量。因此,可以通过检查 Web 图,发现这些社区的特征。从具体的实现细节来分,大概可以分为 3 个不同的实现途径:基于 HITS 的技术、基于有向二分图的技术和基于网络流量的技术。前两个属于依据互连的相互连接的 hub 和 authority 发现社区的算法(即前面的第一类),最后一个属于依据成员链接密度来发现社区的算法(即前面的第二类)。

(1) 基于 HITS 的算法。纯 HITS 算法是 Gibson 等^[2]在 Kleinberg 的 HITS 基础上进行的。将社区定义为一个由“hub”页面连接起来的、很稠密的“authority”页面构成的核。由 HITS 抽取的有序结构的程度越大,(社区)相关页面的数量、超级链接的密度就越大。ARC 和 CLEVER^[4-5]是在纯 HITS 算法的基础上考虑了页面的文本内容。Dean 和 Henzinger^[6]提出的 Companion 算法和 Toyoda 和 Kitsregawa^[7]提出的 Companion 算法也是对 HITS 算法的扩充,不仅考虑链接,而且还考虑链接在页面上的次序。该方法除了对邻接矩阵的边加权考虑之外,重要的改进是在构造邻接子图时与前面的方法不同。

(2) 基于二分有向图的技术。Kumaret 等^[1,3]提出的拖网(trawling)算法和 Reddy 等^[8]提出的放松引用(relaxed-cocitation)都属于这一类型。该方法将社区抽象为一个二分有向图(complete bipartite graph)。稠密的二分图可能包含至少一个社区。和 HITS 不同,其数据来源不是依据某个主题的,而采用的是一般的爬取结果。

(3) 基于流量的技术。Flake 等^[9]根据图形理论,从另外一个角度去提出了发现 Web 社区的方法。将社区定义为在 Web 图中具有这样一些特性的页面的集合,社区内的页面之间的链接(在两个方向)的密度要大于社区之间页面链接的密度。

3 种方法都从各自的角度去发现 Web 社区,并取得了令人鼓舞的成果。我们又将这 3 种方法按照其发现的社区的数据集的源分为两大类:第一类是面向主题的社区发现技术,HITS 算法和流量的技术均属于这一类,其特点是在收集数据集之前,首先要给定某个主题,然后利用现有的搜索引擎的返回结果作为种子集合,在种子集合的基础上构造数据集。

第二类是无主题社区发现技术,拖网算法属于这一类,页面的集合主要来源于网络爬取器(crawler)爬取的页面集合.

因此,第一类方法的结果是具有主观性的,但优点是数据集较小(一般几千个节点),使得应用许多的链接分析算法是可行的.缺点是由于事先确定主题,而只能发现某个主题下的社区,而且带有一定的主观因素.因此我们说社区是主观的.由于是主观的,所以只能发现人们明确定义的社区.

第二类方法可以发现大量的社区,并且发现的社区是客观的,尤其在潜在社区出现时,人们还不知道社区的主题,因此,无主题方法可以发现潜在的社区.缺点是数据量很大,例如,拖网算法的数据集达到 200 百万页面的数量级,因此,许多链接分析方法无法直接应用到该数据集之上.

本文所介绍的研究是属于无主题的方法.但是和前面的方法不同的是在无主题下大量数据集的基础上,采用了流量中的社区特征方法,而算法上应用了 MCL 图形聚类的方法,可以发现客观存在的大量的社区.由于 MCL 是一种快速和可伸缩的高效算法,并且我们将镜像页面的删除过程放在聚类之后完成,可以大大减少比较的代价,加快了社区发现的速度.

3 背景基本知识和 MCL 方法

本小节中,我们先介绍一些图形知识和 MCL 图形聚类算法.

3.1 Web 的图形表示

Web 可以抽象为一个巨大的图.图的节点表

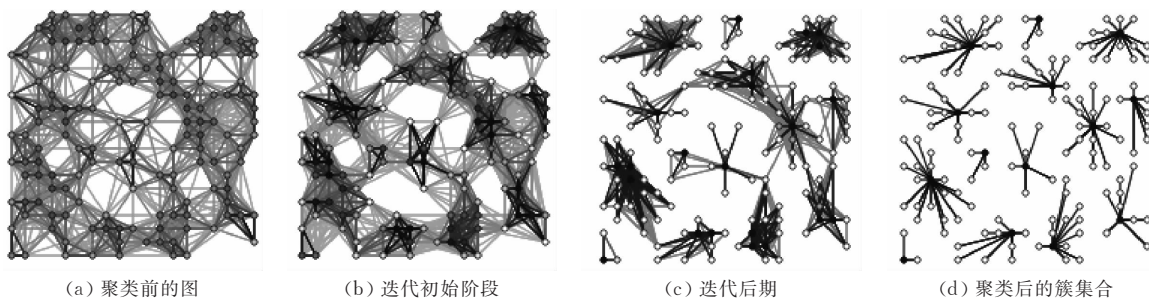


图 1 MCL 的聚类过程

两个节点之间边的灰度级别代表两个方向上的流量.图 1 给出了 MCL 聚类过程的 4 个阶段.从图 1 中可以看出,最终通过流可以分割出不同的区域,结果就使得原图变成了簇的集合.

示 Web 页面,图的边表示超级链接.一个有向图 $G=(V,E)$ 由节点的集合 V 和边的集合 E 组成,边可以表示成为一个有序的节点对.一个无向图 $G=(V,E)$ 由节点的集合 V 和边的集合 E 组成,边可以表示节点之间存在的无向边.

3.2 MCL 图形聚类算法

由 Dongen^[10]提出的 MCL(Markov Cluster)是一种快速的图形聚类算法.MCL 使用关联随机矩阵的简单几何算法,而无需预先了解有哪些可能潜在的簇结构.MCL 利用了记录图中随机漫游聚类结构上抵达的次数.每个节点在各个可能的方向都会有遍历的机会.当大量的漫游者从相同起点开始漫游时,每一个漫游者通常选择不同的路径.该算法的关键思想就是“随机漫游者抵达稠密的簇后,在抵达大部分节点之前不会轻易离开该簇”.MCL 不是模仿实际上的随机漫游,而是不断地修改一个转移概率矩阵.从一个矩阵 $M=M(G)$ (对应一个长度最多为 1 的随机漫游),重复执行下列 2 个操作:

(1) 扩展(expansion). M 取幂 $e \in N > 1$,模拟在当前的转移矩阵上随机漫游走过 e 步.

(2) 膨胀(inflation). M 在抵达第 r 次幂后,重新规范化(re-normalize), $r \in R^+$.

该操作重复执行一直到状态周期变化或者到达一个确定的值.周期为 $k \in N$ 的循环状态表示矩阵在 k 次扩展和膨胀后内容保持不变.而确定值是指周期为 1 的循环状态.文献[10]证明了 MCL 很容易在确定值下结束.因此,通过最终的矩阵,通过图的连接部分可以得到聚类结果.图 1 展示了 MCL 聚类的各个过程.

4 基于图形聚类的社区发现算法

下面我们介绍整个算法的执行过程.首先,需要

一个 Web 页面爬取器(Crawler)获得基本的 Web 页面数据,然后根据每个页面中的链接确定 Web 图的边,这样就形成了一个巨大的 Web 图.在这个 Web 图的基础上,删去不重要的边和节点,形成一个核心图.对核心图使用 MCL 算法,就形成了聚类后的簇集合.在对每个簇进行镜像页面的删除,然后对簇内的所有节点作是否社区成员的判定,最终得到可能的社区候选集.检查这些候选集,就可以发现存在的社区.下面分别介绍各阶段的详细过程.

4.1 收集数据集

数据主要来源于 Web 爬虫(crawler).我们经过选择和比较,认为 Larbin^①是一个不错的 Web 爬虫. Larbin 是一个通用的 Web 爬虫,主要用于搜索引擎数据库数据的获取,如果网络足够快,在一个标准 PC 上可以爬取 100 百万个页面.我们的服务器是 DELL,有 4 个 CPU,1GB 内存,2 个 36GB 硬盘,6 个 146GB 硬盘, Linux 操作系统.考虑到各种情况下的结果要比较,我们对 4 个大硬盘采取不同配置下分别爬取,直至硬盘空间满.每个大硬盘上大约有 3000 个目录,每个目录中包含 2000 个页面文件,每个硬盘大约有 600 万个页面.

4.2 形成边文件

接下来的工作是链接信息的抽取.对于每个页面的内容,我们的抽取原则如下:

(1) 不抽取相对链接(页面内的链接),只抽取具有“http://”开始的 url.并且包含在和之中的 url.

(2) 不抽取动态链接,忽略带参数的 url.

(3) 不抽取超长的 url,我们将长度限制在 100 字符内.

(4) 不重复抽取同一域(站点就是 url 的第一个域)的链接,即 www.edu.cn 和 www.edu.cn/xxx 仅保留前一个 url,而防止统一站点内的链接造成假的社区.

(5) 删除出链接为零的页面.如果出链接页面的数量为零,则没有对应的边.

按照上述规则可以产生边文件.边文件中的每个纪录对应一个页面的 url.为了能很好地记录下整个 Web 图,我们对每条边采用一个源 url 对应多个目的 url 的记录形式.边文件的内容如下:

```
source_url_A
    destin_url_A1
    destin_url_A2
    :
    :
```

```
    destin_url_An
source_url_B
    destin_url_B1
    destin_url_B2
    :
    :
```

由于 Larbin 爬取的结果以 2000 个页面为一组,放在一个目录下.我们就为每一个目录产生一个边文件,这样每个 Web 数据集有大约 3000 个边文件.

4.3 通过边文件产生 Web 图

Web 图采用邻接表的形式记录,即整个 Web 图是边<source_url,destin_url>的集合.由于 url 的长度不统一,也不利于排序.所以我们采用 128 位 md5 指纹函数(fingerprint)来表示每个 url.

由于 Web 图的数据量非常庞大,我们需要采用处理速度较快的数据库.在 Web 数据库的处理过程中,只需要较简单的排序、查找等操作,而主要的问题在于数据库的访问速度.数据库采用了 BerkeleyDB^②,该数据库在处理大量的数据方面速度很快,非常适合建立 Web 图数据库.

由于 Web 图的建立是采用邻接表的方法,因此 BerkeleyDB 数据库中的每个记录只要记录一条边的信息.在 BerkeleyDB 数据库中,每个记录都是一个<key,data>值对.所以,每条边被记录在<key,data>值对之中.其中 key 保存源 url 的 md5 值,data 保存目的 url 的 md5 值.由于对于每个不同的 url 的 md5 可以唯一地标识,因此,在数据库中,每个 url 的 md5 用 source_id 和 destin_id 表示.

为了能够方便地访问 Web 图,我们建立了两个 Web 图数据集.一个是按照<source_id,destin_id>排序的边记录方法建立,由于 key 是有序的,所以我们称之为按照源的边记录数据集,记为 eos.db(edge on source).另一个是按照<destin_id,source_id>排序的边记录方法建立,记为 eod.db(edge on destin).

为了记住 md5 指纹函数对应的 url,我们还需要建立一个数据库,称为 url.db.其中<key,data>值对中,key 保存页面的 md5,data 保存对应 url.

4.4 根据入度和出度确定候选集合

许多研究表明,Web 图的入度、出度的分布符合 Power-law 定理^[11].前期的社区发现研究利用这一定理对数据集进行裁剪,去掉一些对于社区形成

① http://sourceforge.net/project/showfiles.php?group_id=42562

② <http://www.sleepycat.com>

影响不大的节点,同时又要去除一些著名的站点的页面.

对社区形成影响较小的页面是指那些入度和出度都很小的页面.这些页面链接对社区的形成所起到的作用相对较少.而另外一些著名的网站,例如,yahoo!,google 等拥有很大的入度和出度,这些页面形成的密集链接对社区的发现会造成不利的影
响.考虑这样一些因素,我们选择了 $[n,m]$ 区间限定每个页面的入度和出度.和以前的方法不同,在对 Web 图的边裁剪的考虑因素中,我们同时考虑入度和出度.因此,具有一定的入度和出度的页面也具有好的连接其他页面的能力,我们选择具有这样能力的边作为选择社区的候选集合.

因此,入度和出度同时小于 n 和同时大于 m 的页面被裁剪掉.我们对 eos.db 和 eod.db 进行了出入度的裁剪,形成了核心 Web 图的数据集,用 ceos.db 和 ceod.db 表示,定义方法同 eos.db 和 eod.db.

4.5 MCL 对核心数据集聚类

现在可以将 MCL 算法对核心 Web 图做聚类分析.首先我们要根据核心图的数据集产生图形输入矩阵的形式.MCL 可接受边列表的输入形式,即每条边以“ $n1\ n2\ \$$ ”的列表形式建立输入文档.其中 $n1$ 和 $n2$ 代表节点的编号.因此,我们需要为每个节点重新产生节点编号,建立 md5 到节点编号的转换表和反向转换表.反向转换表是用于对计算结果恢复 md5 值的转换.

对 MCL 聚类算法的应用,先对可输入的图形连接表进行转换,然后开始 MCL 聚类计算.MCL 聚类算法可以由多个参数供选择,提供多种条件下的聚类.我们选择了其他缺省参数条件下的聚类,而采用了聚类粒度较小的参数,这样我们希望能聚类得到一些窄主题社区集合,有利于发现潜在的社区.经过聚类后的结果表现为一些页面的簇(cluster).通常规模大的簇排列在前面.我们对簇的规模做了限制,过滤掉了节点数小于 3 的簇.

4.6 从聚类结果中发现社区

当 MCL 聚类算法执行之后,聚类结果是集合 $F = \{C_1, C_2, \dots, C_m\}$, 包含 m 个簇(cluster).每个簇 $C_i (i=1, 2, \dots, m)$ 又是页面的集合.由于图形聚类是按照链接的稠密程度进行的,所以,一个簇内的任意两个页面一定是可达的.

后面的处理分为两个部分,一是镜像页面的删除部分,二是社区成员的确定部分.

(1) 镜像页面删除

Web 中存在大量的镜像页面,其中为提高可靠性或者减轻访问负载而设置的镜像服务器的资源重复外,还有许多大量的复制页面.这些镜像页面会产生一些虚假的社区.因此有必要删除这些镜像页面.

我们采用和文献[7]方法相同的处理方法,即仅根据页面出链接的方法删除镜像页面.镜像页面首先出链接数应该大于 8,如果两个页面之间共享出链接(out-links)的数量比率超过 80%,其中一个页面可以作为镜像页面删除.这里的 8 和 80% 都是根据经验获得的参数.

这个方法的好处是方法简单,容易实现,但理论上存在误删非镜像页面的可能性.从实验中我们发现,所删除的页面基本上都是镜像页面.以后我们将考虑出链接和文本内容结合的方法进一步提高精度.

和其他社区发现方法不同之处在于将镜像页面的删除放在聚类之后进行,不需要对整个 Web 图进行比较,如拖网方法中采用文献[12]的方法删除镜像页面,首先要对整个图的页面进行分类,否则页面的两两比较将是一个天文数字.

然而聚类本身就是将无关的页面分离开来,无关的页面之间不存在链接.因此,镜像页面的删除就只需要在聚类后的每个簇内进行,大大减少了不必要的处理.

(2) 社区成员确定

现在我们得到了一些簇,即一些由页面组成的集合.这些集合是按照链接的稠密程度聚集在一起的.我们还需要对每个簇内的节点进行排序,因为许多簇中的节点数量较大,排序有利于我们寻找重要的节点来确定主要社区的主题.在我们前提假设中社区表现为链接较为紧密的页面集合.因此,如果一个页面属于某一个社区,那么它和社区其他页面之间的链接关系肯定满足一定的条件.我们可以通过检测图形聚类结果簇中的每一个页面和簇其它页面之间的紧密程度来判定该页面是否属于可能的社区.最终删除不符合条件的页面,剩下的页面集合就可能是一个社区.

簇是由页面组成的集合 $S = \{s_1, s_2, \dots, s_m\}$, 其中包含 m 个元素,每个元素是一个页面.我们根据簇内的每个页面和簇内其他页面之间的超级链接的紧密程度来确定该页面在整个簇中的重要程度.

设 $p \in S$,我们用 $I(p, S)$ 表示 p 到 S 中其他元素的超级链接数量, $O(p, S)$ 表示 S 中其他元素到 p 的超级链接数量.

因此,可以定义集合 S 中任何一个元素 p 和 S 的紧密程度.

定义 1. 设 S 为一个页面的集合, p 为 S 中的一个页面,则 p 和 S 的紧密程度为 $Tightness(p, S)$:

$$Tightness(p, S) = (I(p, S) + O(p, S)) / |S| \quad (1)$$

其中, $|S|$ 表示 S 中元素的个数. 我们对簇内的每个元素计算其相对 S 的紧密程度, 然后按照紧密程度对所有元素排序. 我们可以设定一个阈值, $T_{threshold}$, 可以删除紧密程度低于它的页面. 由于一个页面的删除可能会影响其他页面紧密程度的计算, 所以, 一旦发生页面删除, 就需要重新计算所有页面的紧密程度.

判定社区成员的算法流程如下:

```
while(del_flag=ture) {
    delete_flag=false;
    for each  $p \in S$  {
        if  $Tightness(p, S) < T_{threshold}$  {
            delete  $p$ 
            delete_flag=true;
            break;
        }
    }
}
```

通过上述步骤, 我们得到了社区的页面候选集合. 但是到底是否是一个确定的社区, 这依然是一个较为复杂的问题. 和以前的方法相同, 我们依然采取人工分析的方法, 即随机抽取一定数量的簇, 手工分析其内容. 对于较大的簇, 我们可以再次对该簇应用 MCL 算法, 可以发现更加细粒度的簇. 同时, 如何自动地分析簇的页面也是一个具有挑战性的问题. 这将作为我们下一步的研究内容, 我们准备采用基于词频分析统计的方法, 借助人工辅助发现社区的主题.

4.7 社区结果的展现

经过分析后得到社区页面集合表现为一个 Web 子图, 是一系列边的列表, 每个边具有 $\langle source_url, destin_url \rangle$ 的形式. 下面的工作就是如何将这些社区展现出来. 为了更加清晰地展示这些社区以及社区内部各页面之间的关系, 我们采用 Web 图形化的展示工具 TouchGraph^①. TouchGraph 是开放源代码下开发的网络可视化工具, 网络以交互图的形式展现出来. 该图可以实现网络中的导航, 以增大用户的观察视野. 它提供了一套比较新颖的图形化网络部件的管理和组织方法. 可以看到社区内部页面的链接关系.

5 实验和结果

5.1 数据收集和边文件产生

实验是通过对 larbin 爬取配置初始值的设置, 爬取了 4 组不同条件下的数据集. 针对每个配置条件, 开始对一个 146GB 的硬盘装载爬取结果, 直到整个硬盘装满, larbin 停止爬取工作. 最后获取所有的数据集如表 1 所示.

表 1 实验结果

编号	起始 URL	限制的域	Web 图的节点数	Web 图的边数
1	www.ruc.edu.cn	edu.cn	8998440	48484069
2	www.microsoft.com	无限制	7856794	42091102
3	www.edu.cn	.cn	7384744	30122909
4	www.ruc.edu.cn	.edu.cn	5833459	31526509

5.2 结果分析

我们在 4 个数据集上分别应用节点裁减算法, 得到了每个数据集的核心图, 如表 2 所示. 对每个核心图应用 MCL 算法, 就得到每个数据集中包含的簇集合, 如表 2 所示.

发现的社区候选集合数量如表 2 所示.

表 2 获选集合数量

编号	核心 Web 图的节点数	核心 Web 图的边数	发现的社区候选集合数目
1	214533	940702	1742
2	196138	612990	1635
3	44390	478187	2304
4	97163	316642	5217

将 4 个数据集中的簇按照簇的尺寸(簇内节点的数量)和各个尺寸下簇的数量的分布情况用图 2 表示, 可以看出 4 个数据集均满足 Power-law 定理.

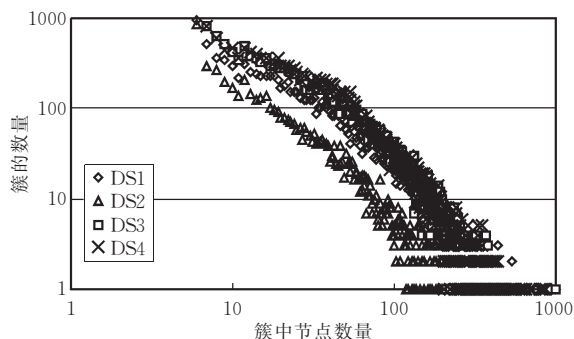


图 2 MCL 簇的分布图

不同的数据集的起始参数不同, 所发现的社区

① <http://www.touchgraph.com/>

候选集合的数目也大不相同. 我们针对数据集 4 的候选集合进行了分析, 手工检查了这些社区页面内容, 发现其中许多集合的页面是围绕某个主题的, 因此, 可以断定是一个 Web 社区. 由于候选集合的数量非常的庞大, 而且对于这些页面的分析工作量也很大. 人工分析是无法满足对整个候选集合的分析. 因此, 我们寻找出一些具有明确主题意义的社区. 图 3~图 5 是通过 TouchGraph 展示的 3 个社区.

5.3 结果说明

我们挑选了 4 个数据集, 每个集合的规模大约

为 146GB 左右. 每个数据集合的产生在 Dell 服务器上运行 2 周. 由于我们对原始的 Web 图进行了边的筛选处理, 就是删去对社区产生影响不大的边, 使得 Web 图的规模大大减小. 从表 1 和表 2 的对比中可以看到这一点. 最终得到的社区候选集合数量是很大的. 从图 3、图 4 来看, 每个社区内部页面链接的结构也是大不相同的. 因此, 还需要进一步地对社区内部链接结构信息的挖掘展开研究. 此外, 社区之间的链接关系也需要深入研究.



图 3 包含“汽车保险”相关主题的社区



图 4 包含“律师”相关主题的社区



图 5 包含“汽车杂志”相关主题的社区

并不是所有的候选集合都具有明确的主题, 有些集合的主题不是很明确, 有些集合甚至包含一个以上的主题. 在这里, 我们通过手工检查的方法发现

大多数社区是具有语义关联的页面集合. 但是, 要想手工检查到所有的社区依然是不现实的. 因此, 如何通过自动的方法去检查这些社区的主题也是一个值得研究的问题.

6 结论和未来的工作

本文中我们介绍了 Web 社区的研究现状和社区发现算法. 在图形理论的基础上, 描述了基于 MCL 图形聚类的方法. 和以前的社区发现如 HIST 算法、二分核算法和基于流量算法不同, 本文通过图

形聚类 MCL 对 Web 图进行簇的分割,将簇内链接紧密的页面集合聚为一类,而簇间的链接较为稀疏.在聚类后簇集合的基础上进行镜像页面的删除可以大大减少比较的代价.最后,根据簇内每个元素和整个簇的紧密关系,确定该元素是否为社区成员.这些簇作为可能存在社区的页面集合,我们称为社区候选集合.分析这些候选页面集合可以发现许多潜在的社区.最终的实验结果表明,本方法可以有效地发现 Web 中存在的许多社区.

采用 MCL 的研究刚刚开始,虽然可以发现许多社区,依然存在许多需要进一步研究的问题.未来的研究工作可包含这样几个方面:(1)从 MCL 簇抽取的社区还缺乏一个真实性和可靠性的评价手段,目前仅依靠人工方法是远远不够的.因此,可以考虑结合文本内容的页面相似度的判定方法;(2)社区层次结构的抽取,包括社区内层次和社区间层次的抽取;(3)社区信息的管理和应用.如何有效地管理这些社区,包括对这些社区信息的维护、更新、评价等以及如何将社区信息应用于 Web 搜索技术之中也是我们下一步开展的研究.

参 考 文 献

- [1] Kumar R, Raghavan P, Rajagopalan S et al. Trawling the Web for emerging cyber-communities//Proceedings of the 8th International WWW Conference. Toronto, Canada, 1999; 403-415
- [2] Gibson D, Kleinberg J, Raghavan P. Inferring Web communities from link topology//Proceedings of the 9th ACM Conference on Hypertext and Hypermedia. Pittsburgh, PA, USA, 1998; 225-234
- [3] Kumar R, Raghavan P, Rajagopalan S et al. Extracting large-scale knowledge base from the Web//Proceedings of the 25th International Conference on Very Large Data Bases (VLDB'99). Edinburgh, Scotland, 1999; 639-650
- [4] Chakrabarti S, Dom B E, Raghavan P et al. Automatic resource compilation by analyzing hyperlink structure and associated text. Computer Networks and ISDN Systems, 1998, 30(1-7): 65-74
- [5] Chakrabarti S, Dom B E, Kumar R et al. Mining the Web's link structure. IEEE Computer, 1999, 32(8): 60-67
- [6] Dean J, Henzinger M R. Finding related pages in the world wide Web//Proceedings of the 8th International WWW Conference. Toronto, Canada, 1999; 389-401
- [7] Toyoda M, Kitsuregawa M. A Web community chart for navigating related communities//Proceedings of the 10th International WWW Conference. Hong Kong, China, 2001; 62-63
- [8] Reddy P K, Kitsuregawa M. Inferring Web communities through relaxed-cocitation and power-law. Tokyo, Japan; Kitsuregawa Lab, Annual Report, 2001
- [9] Flake G W, Lawrence S, Giles C L. Efficient identification of Web communities//Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston, MA, USA, 2000; 150-160
- [10] Dongen S V. Graph clustering by flow simulation [Ph. D. dissertation]. University of Utrecht, 2000
- [11] Broder A, Kumar R et al. Graph structure in the Web. Computer Networks, 2000, 33(1-6): 309-320
- [12] Broder A Z, Glassman S C, Manasse M S et al. Syntactic clustering of the Web. Computer Networks, 1997, 29(8-13): 1157-1166
- [1] Kumar R, Raghavan P, Rajagopalan S et al. Trawling the Web for emerging cyber-communities//Proceedings of the 8th International WWW Conference. Toronto, Canada, 1999; 403-415
- [2] Gibson D, Kleinberg J, Raghavan P. Inferring Web communities from link topology//Proceedings of the 9th ACM Conference on Hypertext and Hypermedia. Pittsburgh, PA, USA, 1998; 225-234
- [3] Kumar R, Raghavan P, Rajagopalan S et al. Extracting large-scale knowledge base from the Web//Proceedings of the 25th International Conference on Very Large Data Bases (VLDB'99). Edinburgh, Scotland, 1999; 639-650
- [4] Chakrabarti S, Dom B E, Raghavan P et al. Automatic resource compilation by analyzing hyperlink structure and associated text. Computer Networks and ISDN Systems, 1998, 30(1-7): 65-74
- [5] Chakrabarti S, Dom B E, Kumar R et al. Mining the Web's link structure. IEEE Computer, 1999, 32(8): 60-67
- [6] Dean J, Henzinger M R. Finding related pages in the world wide Web//Proceedings of the 8th International WWW Conference. Toronto, Canada, 1999; 389-401
- [7] Toyoda M, Kitsuregawa M. A Web community chart for navigating related communities//Proceedings of the 10th International WWW Conference. Hong Kong, China, 2001; 62-63
- [8] Reddy P K, Kitsuregawa M. Inferring Web communities through relaxed-cocitation and power-law. Tokyo, Japan; Kitsuregawa Lab, Annual Report, 2001
- [9] Flake G W, Lawrence S, Giles C L. Efficient identification of Web communities//Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston, MA, USA, 2000; 150-160
- [10] Dongen S V. Graph clustering by flow simulation [Ph. D. dissertation]. University of Utrecht, 2000
- [11] Broder A, Kumar R et al. Graph structure in the Web. Computer Networks, 2000, 33(1-6): 309-320
- [12] Broder A Z, Glassman S C, Manasse M S et al. Syntactic clustering of the Web. Computer Networks, 1997, 29(8-13): 1157-1166



YANG Nan, born in 1962, associate professor. His research interests include Web mining, knowledge discovery in Web, graph clustering and information retrieval.

LIN Song-Xiang, born in 1981, M. S. candidate. His

research interests include Web communities and graph clustering algorithm.

GAO Qiang, born in 1981, M. S. candidate. His research interests include Web communities, Web crawler.

MENG Xiao-Feng, born in 1964, professor, Ph. D. supervisor. His research interests include database system, Web data integration, XML data management and mobile database.

Background

Web community is an important social activity in the evolution of Web. Many communities have been aware of by people. But there are many communities implicit to people and how to mining them is a hard job. Although many researches on communities have gained great progress, but there are still many problems unsolved.

This work is part of 211 projects of ministry of education, entitled Research on Discovery Technology of Web Resources. In this paper authors analyzes typical algorithms of

Web communities' discovery in the present. Under the ground of non-topic pre-defined and implicit communities, they propose a new method. They combine both characteristic structure of community and the clusters of Markov Graph Clustering(MCL) to find implicit communities. This work begins to apply graph clustering technology to very large scale graph and gain progress. The next work will focused on mining the hierarchical structure of Web and communities and automatic topic extraction on clusters of Web.