基于 Web-Log Mining 的 Web 文档聚类*

苏 中[1,2], 马少平[1,2], 杨 强[3], 张宏江[4]

[1](清华大学 计算机科学与技术系,北京 100084);
[2](清华大学 智能技术与系统国家重点实验室,北京 100084);
[3](Simon Fraser 大学,加拿大);
[4](微软中国研究院,北京 100080)
E-mail: suzhong_bj@hotmail.com
http://www.tsinghua.edu.cn

摘要: 速度和效果是聚类算法面临的两大问题.DBSCAN(density based spatial clustering of applications with noise)是典型的基于密度的一种聚类方法,对于大型数据库的聚类实验显示了它在速度上的优越性.提出了一种基于密度的递归聚类算法(recursive density based clustering algorithm,简称 RDBC),此算法可以智能地、动态地修改其密度参数.RDBC 是基于 DBSCAN 的一种改进算法,其运算复杂度和 DBSCAN 相同.通过在 Web 文档上的聚类实验,结果表明,RDBC 不但保留了 DBSCAN 高速度的优点,而且聚类效果大大优于 DBSCAN.

关 键 词: 数据库;聚类;Web mining;数据挖掘
中图法分类号: TP311 文献标识码: A

数据挖掘就是试图在大型数据库中发现隐含模式的过程.聚类算法是数据挖掘中的一个重要的分析工具.作为统计分析的一个分支,聚类分析在过去的 40 多年中得到了深入的研究,并广泛运用于许多应用领域.对于数据挖掘的任务,聚类分析的诱人之处是有可能在不需要知道任何数据的层次信息的前提下,从数据中发现层次或类的关系.然而,在数据挖掘和机器学习中运用的聚类分析并没有得到非常成功的运用.其原因就是在大规模数据库中,通常算法的速度和效果不能令人满意.

现在已有许多聚类算法,例如 K-means[1],HAC(hierarchical agglomerative clustering)[2~4],CLANRNS (clustering large applications based on randomized search)[5]等,这些方法都是面向小数据集合的,不太适合数据挖掘所面对的大型数据库.

DBSCAN(density based spatial clustering of applications with noise)[6]是一种通过对局部密度分析,将相邻点聚集在一起的聚类算法.在整个算法进行过程中,它只对数据库进行一次扫描.如果 DBMS 对相邻点的查询效率很高(DBMS 的查询效率目前已经完全满足这一条件),DBSCAN 的效率将非常令人满意.它是当前面向大数据集聚类算法中最快的一种.然而,由于算法本身在整个聚类过程中使用固定的参数(这将在下一节中讨论),使得对于真实环境数据集的聚类,往往其聚类的效果不好.其主要原因是,由于其定义的密度的传递性质,往往将绝大多数的数据点都聚集在非常少的几类中(通常是一类).

在本文中,我们提出了一种基于密度的递归聚类算法 RDBC(recursive density based clustering algorithm).此算法可以智能地、动态地修改其密度参数.RDBC 是基于 DBSCAN 的一种改进算法.算法的基本思想是,我们并

* 收稿日期: 2000-04-03; 修改日期: 2000-07-20
基金项目: 国家重点基础研究发展规划 973 资助项目(G1998030509)

作者简介: 苏中(1976－),男,上海人,博士生,主要研究领域为基于内容图像检索,模式识别,网络数据挖掘;马少平(1961－),男,河北唐山人,博士,教授,博士生导师,主要研究领域为模式识别,信息检索,网络数据挖掘;杨强(1961－),男,北京人,博士,教授,主要研究领域为机器学习,数据挖掘,知识系统;张宏江(1960－),男,黑龙江哈尔滨人,博士,研究员,主要研究领域为视频和图像内容分析与检索,计算机视觉,信息系统.

不对原始数据集进行聚类,而是通过从数据集合中抽取高密度点生成新的数据集合,并修改密度参数,反复进行这一过程,直到生成的数据集合可以很容易地被聚类为止,然后以此结果为基础,再将其他点逐层地吸附到各个类中.RDBC 的运算复杂度和 DBSCAN 相同.通过对 Web 文档数据的聚类实验,结果表明,RDBC 不但保留了 DBSCAN 高速度的优点,而且聚类效果大大优于 DBSCAN.

本文第 1 节给出了一些相关工作.第 2 节描述了 RDBC 算法.第 3 节讲述了运用 Web 日志文件进行文档聚类的方法.第 4 节是实验描述.第 5 节是总结.

1 相关工作

基于密度的聚类方法是一类典型的聚类算法.其基本思想来源于将密度大的邻近点聚集在一起.这样,每一类就是一个高密度的点区域.因此,它不需要用户定义类个数.DBSCAN 是由 Ester 等人在 1996 年引入的一种一次扫描数据库的基于密度的聚类算法.如果 DBMS 对相邻点的查询效率很高(DBMS 的查询效率目前已经完全满足这一条件),DBSCAN 的效率将非常令人满意.它是当前面向大数据集聚类算法中最快的一种.同时,作为一种基于密度的聚类方法,它可以处理任意形状的聚类问题.

下面详细介绍一下 DBSCAN 算法.DBSCAN 有两个参数:半径 ε——作为密度计算的距离表示;数值 Minpts——密集点所必需的在半径 ε 内拥有的最少的其他点的数目.通过这两个参数我们就可以计算在任何点周围的密度值.DBSCAN 就是在 N 维空间中反复计算各点的密度,并将它们按照密度聚集成类.下面给出一些必要的定义.

定义 **1(ε-相邻点)**. 对于点 p 的 ε-相邻点,表示为 $N_\varepsilon(p)$,其中 $N_\varepsilon(p)=\{q\in D|dist(p,q)\leq\varepsilon\}$.

对于给定的值 Minpts,点 q 如果是点 p 的 ε-neighborhood 点,则称点 p 从点 q 直接密度-到达.

定义 **2(直接密度-到达)**. 对于给定的 ε 和 Minpts,点 p 从点 q 直接密度-到达,当且仅当

(1) $p\in N_\varepsilon(q)$;

(2) $|N_\varepsilon(q)|\geq MinPts$(密集点条件).

在第(2)种条件下,q 满足密集点条件,因为周围有足够的点围绕着它.

运用直接密度-到达的定义,我们可以通过传递关系定义出密度-到达.

定义 **3(密度-到达)**. 对于给定的 ε 和 Minpts,点 p 从点 q 处密度-到达,当且仅当存在这样一个点的序列:$p_1,\ldots,p_n,p_1=q,p_n=p$,其中点 p_{i+1} 从点 p_i 处直接密度-到达.

定义 **4(密度-连接)**. 对于给定的 ε 和 Minpts,点 p 和点 q 密度-连接,当且仅当存在这样一个点 o,点 p 和点 q 均与点 o 密度-到达.

定义 **5(类)**. 假定 D 是具有距离定义的数据库.对于给定的 ε 和 Minpts,某一类 C 就是满足以下条件的非空子集:

(1) $\forall p,q$,如果 $p\in C$ 并且对于给定的 ε 和 Minpts,点 q 从点 p 处密度-到达,那么 $q\in C$.

(2) $\forall p,q\in C$,对于给定的 ε 和 Minpts,点 p 和点 q 密度-连接.

基于上述类的定义,对于给定的 ε 和 Minpts,DBSCAN 算法从任意一个密集点出发,反复地按照定义 5 的要求扩展该类.为了支持磁盘处理,所有被标上属于某一类的点将不会再对其进行计算.因此,算法的计算时间复杂度为 $N^*\log(N)$.然而,由于算法本身在整个聚类过程中使用固定的参数 ε 和 MinPts(Ester 提供了一种自动求得参数的方法),使得对于真实环境数据集的聚类,往往其聚类的效果不好.主要一点原因就是由于其定义的密度的传递性质,往往将绝大多数的数据点都聚集在非常少的几类中(通常是一类).

下面给出算法的描述.

算法 **1**. Algorithm DBSCAN (DB,ε,MinPts)

(1) for each o, DB do

(2) if o is not yet assigned to a cluster then

(3) if o is a core-object then

(4) collect all objects density-reachable from o according to ε

(5) and MinPts;

assign them to a new cluster

2 RDBC 算法

DBSCAN 是一种通过对局部进行密度分析,将相邻点聚集在一起的聚类算法.在整个算法过程中,它只对数据库进行一次扫描.由于算法本身在整个聚类过程中使用固定的参数,使得对于真实环境数据集的聚类,往往其聚类的效果不好.其主要的一点原因就是由于其定义的密度的传递性质,往往将绝大多数的数据点都聚集在非常少的几类中(通常是一类).例如,图 1 所示,数据之间严重粘连,在现实数据中这是普遍存在的.DBSCAN 将把绝大多数的点聚集为一类.而从图中来看,这种聚类显然不具有合理性.RDBC 就是希望避免这种情况的发生,它通过从数据集合中反复抽取高密度点生成新的数据集合,直到生成的数据集合可以很容易地被聚类为止,并在这个数据集合上给出各个类的初始分割.

(a) Initial data points distribution (b) Data distribution after two iterations of core-point extraction processes. It can be clustered into three clusters easily

(a) 初始数据点分布 (b) 经过两次核心点抽取过程后的数据分布,可以看到数据很容易被聚到3个类中

Fig.1 Brief description of the idea of the algorithm

图 1 算法思路的简单描述

RDBC 的算法流程是这样的:首先设置初始密度参数ε和 Mpts,设置方法参见文献[6],利用绘制降序 k-距离图来确定.以下给出我们的算法的详细流程说明.

算法 **2**. Algorithm RDBC (WebPageSet)

(1) Set the initial value of Eps_1, and the Mpts_1.

(2) $\varepsilon=\varepsilon$_1,Mpts=Mpts_1;

(3) WebPageSet=Web_Log;

RDBC (ε,Mpts,WebPageSet)

{

(4) Use ε and Mpts to get the core points set Cset

(5) if size(CSet)>size(WebPageSet)/2

{// Stopping criterion is met.

(6) DBSCAN(WebPageSet,ε,Mpts);

}

(7) else

{// continue to abstract core points from L

(8) Update ε and Mpts according to Cset

(9) RDBC(ε,Mpts,CSet);

(10) Collect all other points in (WebPageSet-CSet) around clusters found

(11) in Step (10) according to ε*2.

}
}

DBSCAN 算法的时间复杂度是 $O(N*\log N)$.RDBC 算法进行了有限次的抽取密集点的过程、一次 DBSCAN 和有限次的吸附过程,因此其算法复杂度为 $O(c*\log N+N*\log N+L)=O(N*\log N)$.因此,在时间复杂度上,RDBC 和 DBSCAN 完全一样.而且由于 RDBC 最终作聚类的数据集只是原始集合的一个小子集,所以在实验中,其实际速度比 DBSCAN 还要快一些.

3 在 Web 日志文件上的文档聚类

我们的聚类过程可以分为以下 4 步:

Step 1. 对 Web 日志文件的处理.

(1) 在日志文件中清除由搜索引擎的 Crawler 以及 Proxy 发出的 Web 申请,并将其余数据装入数据库.删除日志中的图片申请,因为通常这些图片都包含在某个页面中,对这些图片的申请是由 HTTP 协议发出的,而不是由用户发出.

(2) 抽出该数据库中所有的对话过程.对于一个用户的申请,如果相邻两个 Web 申请的时间间隔大于某个域值 T 的话,就认为它们属于不同的会话过程.通过实验观察,我们将时间域值定为两小时.

Step 2. 计算在滑动窗口大小内的不同 Web 页面间的申请同发次数.建立页面间的距离矩阵.

(1) 设置滑动窗口的大小.所谓滑动窗口,是指在同一个对话过程中,滑动窗口内的任何两个页面(P_i,P_j)申请被认为是关联的.相反,如果两个页面的申请相隔太远,则认为这两个页面不相关,不记录它们的同发申请.

(2) 统计所有的对话过程,计算出任意两对页面(P_i,P_j)之间的同发申请的次数 $N_{i,j}$.同时,也计算出每一页面单独的申请次数 N_i,N_j.

(3) 计算 $P(P_i|P_j)=N_{i,j}/N_j$.

(4) 计算出页面间的距离矩阵.距离的定义有以下 3 种:

$$Dis1(A,B)=\mathrm{Max}(1/P(A|B),1/P(B|A)), \tag{1}$$

$$Dis2(A,B)=0.5(1/P(A|B)+1/P(B|A)), \tag{2}$$

$$Dis3(A,B)=\sqrt{(1/P(A|B)\cdot 1/P(B|A))}. \tag{3}$$

对于任意一种定义,只要满足 AB 间和 BA 间距离相同,就可以作为距离定义.第(1)个距离的定义来源于 Perkowitz 和 Etzioni 的工作中.后两个距离定义考虑了页面间的相互关系,分别利用了算术和几何平均值.通过实验,我们发现第(3)个距离定义其结果要好于前两种定义给出的结果.

Step 3. 在距离矩阵上运行 RDBC 算法.

Step 4. 输出聚类结果.

4 实验结果

首先介绍一下我们使用的数据文件.数据来源于肯尼迪航天中心(NASA)的 web 服务器.它记录了从 1995 年 8 月 1 日 00:00:00 到 8 月 31 日 23:59:59 所有的访问信息.共有 18 688 个不同的访问 IP 对 15 429 个页面进行了 1 569 898 次 web 访问.我们从中共抽出了 171 529 次有效的会话过程.

我们在上述 3 个数据文件中分别用 RDBC 和 DBSCAN 方法进行了聚类,并对结果进行了比较.表 1 给出的是使用 RDBC 算法聚类的部分结果.如果使用 DBSCAN,那么表中的所有页面都将被聚在同一个类中.通过文件名,我们可以清楚地看到这些类之间确实存在显著的差异,而同一类则是完全相关的.

表 2 和图 2 中显示的是使用 RDBC 和 DBSCAN 聚类结果的比较.从结果中我们可以清楚地看到,RDBC 聚类的速度稍快一些,而且聚类的结果分布更加均匀.如果运用 DBSCAN,那么几乎所有的页面都被聚集到一个类当中.通过对结果的观察,我们发现,RDBC 的聚类结果是合理的,相似页面被聚集在一起.

Table 1 Partial clustering result using RDBC

表 1 运用 RDBC 算法以后的聚类结果

Class① 1	/shuttle/missions/41-c/news/ /shuttle/missions/61-b/ /shuttle/missions/sts-34/ /shuttle/missions/41-c/images/ …
Class 2	/history/apollo/sa-2/news/ /history/apollo/sa-2/images/ /history/apollo/sa-1/sounds/ /history/apollo/sa-9/sa-9-info.html …
Class 3	/software/winvn/userguide/3_3_2.htm /software/winvn/userguide/3_3_3.htm /software/winvn/userguide/3_8_1.htm /software/winvn/userguide/3_8_2.htm …
…	…

①类.

Table 2 Comparison between the clustering result of RDBC and that of DBSCAN

表 2 比较 RDBC 和 DBSCAN 的聚类结果

	RDBC	DBSCAN
Number of pages①	15 429	15 429
Run time② (s)	21	25
ε/Mpts	10/20 5/5	10/20
Number of clusters③	44	4

①页面数,②运行时间,③类个数.

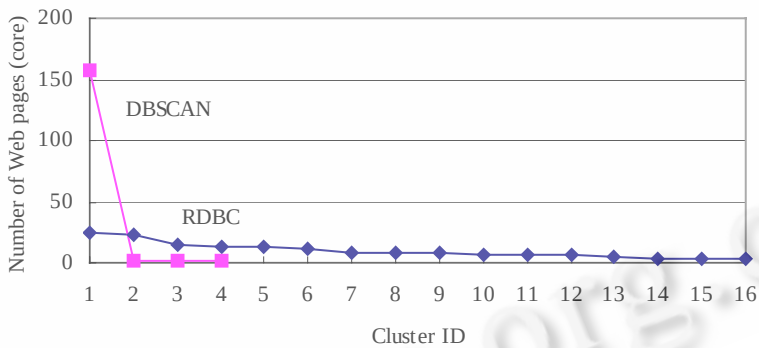

Fig. 2 Comparison of RDBC and DBSCAN

图 2 RDBC 和 DBSCAN 的比较

5 小 结

我们提出了一种基于密度的递归聚类算法(RDBC).此算法可以智能地、动态地修改其密度参数.RDBC 是基于 DBSCAN 的一种改进算法,其运算复杂度与 DBSCAN 相同.通过在 3 个真实数据文件上的对比实验,我们发现,RDBC 不但保留了 DBSCAN 的所有优点,而且聚类效果大大优于 DBSCAN.同时,我们提出了一种利用 web 日志文件对网页进行聚类的有效方法.这些工作正在有关动态网站的研究中进行.

致谢 我们感谢徐晓伟(DBSCAN 的提出者之一)在初始阶段的工作,也感谢 YuHen Hu 的讨论和帮助.另外,David Abrecht,Ingrid Zukerman 以及 MSR Web Support Group 的 Steven Johnson 共享给我们珍贵的数据文件,在此,我们谨表谢意.

References:

[1] Kaufman, L., Rousseeuw, P. J. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.

[2] Sibson, R. SLINK: an optimally efficient algorithm for the single-link cluster method. The Computer Journal, 1973,16(1):20~34.

[3] Bouguettaya, A. On-Line clustering. IEEE Transactions on Knowledge and Data Engineering. 1996,8(2):333~339.

[4] Voorhees, E.M. Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. Information Processing and Management, 1986,22:465~476.

[5] Ng, R., Han, J. Efficient and effective clustering methods for data mining. In: Bocca, J.B., Jarke, M., Zaniolo, C., eds. Proceedings of the 1994 International Conference on Very Large Data Bases (VLDB'94). Santiago, Chile: Morgan Kaufmann, 1994. 144~155.

[6] Ester, M., Kriegal, H.P, Sander, J. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, Evangelos, Han, Jia-wei, Fayyad, U.M., eds. KDD'96——Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. AAAI Press, 1996.

Document Clustering Based on Web-Log Mining*

SU Zhong[1,2], MA Shao-ping[1,2], YANG Qiang[3], ZHANG Hong-jiang[4]

[1](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*);

[2](*State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing* 100084, *China*);

[3](*Simon Fraser University, Canada*);

[4](*Microsoft Research China, Beijing* 100080, *China*)

E-mail: suzhong_bj@hotmail.com

http://www.tsinghua.edu.cn

Abstract: The effectiveness and efficiency are two problems in clustering algorithms. DBSCAN is a typical density based clustering algorithm that is very efficient on large databases. In this paper, a recursive density based clustering algorithm that can adaptively change its parameters intelligently is presented. This clustering algorithm RDBC (recursive density based clustering algorithm) is based on DBSCAN. It can be shown that RDBC require the same time complexity as that of the DBSCAN algorithm. In addition, it is proved both analytically and experimentally that this method yields results more superior than that of DBSCAN.

Key words: databases; clustering; web mining; data mining

* Received April 3, 2000; accepted July 20, 2000
Supported by the National Grand Fundamental Research 973 Program of China under Grant No.G1998030509