ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.3, March 2006, pp.379−387
DOI: 10.1360/jos170379

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于数据流的任意形状聚类算法*

朱蔚恒+, 印 鉴, 谢益煌

(中山大学 计算机科学系,广东 广州 510275)

Arbitrary Shape Cluster Algorithm for Clustering Data Stream

ZHU Wei-Heng, YIN Jian, XIE Yi-Huang

(Department of Computer Science, Sun Yat-Sen University, Guangzhou 510275, China)

+ Corresponding author: E-mail: gz_zwh@263.net, http://www.sysu.edu.cn

Zhu WH, Yin J, Xie YH. Arbitrary shape cluster algorithm for clustering data stream. *Journal of Software*, 2006,17(3):379−387. http://www.jos.org.cn/1000-9825/17/379.htm

Abstract: CluStream is a popular data stream cluster algorithm, however, it is not capable enough to cluster arbitrary shapes and make clusters in periodic data. This paper introduces a new algorithm ACluStream to solve these problems. The ACluStream is based on the partition and assemble of the space and cluster by density. In the experiment, it is shown that ACluStream is better than CluStream in speed and accuracy.
Key words: data stream; clustering; data mining

摘 要: 详细分析了数据流聚类算法 CluStream 的不足之处,如对非球形的聚类效果不好、对周期性数据的聚类变化反映不完整等,并针对这些不足之处提出了一种采用空间分割、组合以及按密度聚类的算法 ACluStream. 实验结果表明,ACluStream 在准确度和速度上都比 CluStream 有较大的提高.
关键词: 数据流;聚类;数据挖掘
中图法分类号: TP311 文献标识码: A

近年来,由于硬件技术的高速发展,人们获取数据的能力得到了极大的提高.现实生活中,经常可以看到这样的情况:大量需要处理的数据以很快的速度产生.例如,美国一条高速公路上的传感器网络每天可以收集到高达几百万条的数据,而电讯电话公司大型交换机上每天记录的通话记录就高达几千万条.由于数据量太大、数据产生的速度太快,按传统的数据库应用模式处理这些数据,即完整、详细地收集这些数据,清洗后将其储存在数据库中,再交由计算机仔细处理已成为不可能完成的任务.由有限的数据到有限的数据处理能力,计算机工作者们面临着新的挑战.

* Supported by the National Natural Science Foundation of China under Grant No.60573097 (国家自然科学基金); the Research Foundation of National Science and Technology Plan Project of China under Grant No.2004BA721A02 (国家科技计划); the Research Foundation of Disciplines Leading to Doctorate Degree of Chinese Universities under Grant No.20050558017 (高等学校博士学科点专项科研基金); the Natural Science Foundation of Guangdong Province of China under Grant Nos.05200302, 04300462 (广东省自然科学基金); the Research Foundation of Science and Technology Plan Project in Guangdong Province of China under Grant No.2005B10101032 (广东省科技计划项目)

Received 2004-09-28; Accepted 2005-03-11

针对如何分析管理这种大量快速的数据问题,人们提出了一类新型应用作为解决方案.这些方案最大的特点是,待处理的数据不再静态、固定地存储在可多次、随机访问的介质中,而是以一种动态、流式的形式出现(并称其为数据流,data stream),对数据只能是顺序的、一次或有限次的访问[1].在最近的研究中,数据流已逐渐成为新一代计算理论与应用的研究热点之一.

与传统的数据库应用相比,应用于数据流的算法有几个明显的特点:首先,由于数据流的速度很快,对算法的响应要求很高,所以数据流算法经常采用用精度换时间的方法,尽量在对数据的一次访问中获得较优的解.一般来说,数据流算法是不可回溯的;其次,数据流算法有很多特点,一些数据库应用中常用的操作在数据流中都是不可行的.如,Sort,Max,Count 等 Blocking 操作[2].

目前对数据流的研究主要集中在以下几个方面:对数据流工作模型的建模、对数据流查询的响应、如何管理数据流、如何对数据流进行挖掘等.在数据流挖掘方面,如何在数据流中进行有效聚类,是一个吸引了研究者很大注意力的问题[3−6].

2000 年,Guha 提出针对数据流聚类的 LOCALSEARCH[5]算法.算法的基本思想是基于分治的思想使用一个不断的迭代过程实现有限空间对数据流进行 k-means 聚类.O'Callaghan 发展了 LOCALSEARCH 的思想.他于 2002 年提出了 STREAM[6]算法,并利用 SSQ 证明了在多种情况下,STREAM 算法的聚类效果都比 BRICH[7]算法要好.但 LOCALSEARCH 和 STREAM 方法都有一个缺点,即只能提供对当前数据流的一种描述,而不能反映流数据的变化情况[4].

2003 年,Barbard 总结了数据流聚类算法的要求[3],并对一些可能适用于数据流的聚类算法做了一次总结.他认为在数据流中聚类要满足 3 个要求:(1) 压缩的表达;(2) 快速、增长地处理新的数据点;(3) 快速、清晰地判断独异点.

CluStream[4]是 Aggarwal 在 2003 年提出的一个解决数据流聚类问题的框架.它使用了两个过程来处理数据流聚类问题:首先,使用一个在线的 micro-cluster 过程对数据流进行初级聚类,并按一定的时间跨度将 micro-cluster 的结果按一种称为 pyramid time frame 的结构储存下来.同时,使用另一个离线的 macro-cluster 过程,根据用户的具体要求对 micro-cluster 聚类的结果进行再分析.

本文第 1 节分析 CluStream 算法存在的一些问题,并简要介绍我们提出的新算法 ACluStream 所使用到的基本概念和一些关于 ACluStream 的理论证明.第 2 节给出一个完整的 ACluStream 算法,详细解析算法的思想和执行过程.第 3 节给出 ACluStream 与 CluStream 具体的实验比较,可以看到与 CluStream 相比,ACluStream 具有很强的空间表达能力.最后,讨论 ACluStream 的不足以及继续研究的方向.

1 问题分析与算法概述

1.1 CluStream存在的问题

CluStream 提供了一个解决数据流中聚类问题的框架,虽然该框架非常优秀,但在这个框架中仍有许多可以改进的地方.

CluStream 使用了一个在线的 micro-cluster 过程对数据流进行初级描述,后续工作都是建立在该过程的输出之上.显然,这个过程是整个聚类算法的基础,但 CluStream 所使用的 micro-cluster 过程的存在一些不足.

micro-cluster 过程使用的是类似 BRICH 算法所使用的聚类特征值来记录它所产生的子聚类.BRICH 算法的缺点必然也带入 micro-cluster 过程中.BRICH 算法记录聚类特征值的方法对球型的聚类效果好,但对其他形状的聚类,BRICH 不能很好地工作[8].所以对非球形的聚类,micro-cluster 过程同样不能给出一个很好的描述.由于 micro-cluster 过程得到的结果是对数据流进行一个初步的描述,所以,如果这个描述本身不精确,那么后续的 macro-cluster 过程将不能得到一个较好的结果.

再观察 micro-cluster 对新数据的处理:它利用数据与最近子聚类中心的距离以及一个预定的距离阈值来判断数据是否属于该子聚类.但距离某一聚类中心近的点不一定就属于该聚类.观察如图 1 所示的例子.

图1中的点集属于两个不同的类别,使用micro-cluster过程进行聚类时,由于各数据点生成次序的不同,最后得到的聚类结果可能有多种.其中一种就是生成图 1 中使用圆圈标出的两个子聚类.显然,这两个子聚类并不能准确区分两个类别,而且它们所覆盖的面积比原来点集的面积几乎大了一倍.此方法在处理流数据时还会造成各子聚类所覆盖的区间重叠,同一位置的点被归入不同的子聚类的情况.

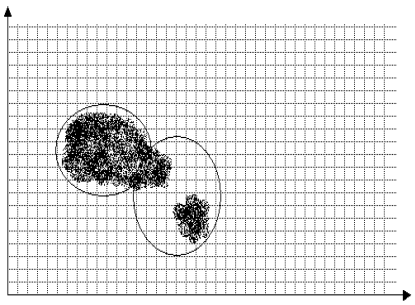


Fig.1 A bad cluster

图 1 一个不好的聚类

更进一步地,micro-cluster 判断一个新来的点是否属于某子聚类的方法是根据该点与子聚类中心的距离.因为子聚类没有严格的空间定义,它的中心随着每一个点的到来而发生改变.所以,不能保证生成子聚类的点最后都包含在一个与子聚类中心距离不大于 r 的圆形邻域中.可以证明,在最坏情况下,子聚类初始部分的点与子聚类中心的距离可能是无限远.

针对上述问题,本文提出了基于密度与空间的 ACluStream(arbitrary-shape CluStream)聚类算法.算法的目标是在对数据流进行初步聚类的同时,尽量保留数据的空间特性.ACluStream 算法采用了与 CluStream 相同的架构:采用两个过程处理数据,一个在线的过程给出对当前数据流特征的描述,按pyramid frame的结构储存这些描述;一个离线的过程利用这些描述处理查询.

1.2 ACluStream使用的基本概念

定义 n 维空间 S 上点集 D.D 中的元素以数据流的形式到达.算法的目标是按到达的顺序得到 D 中某一批数据的一个聚类描述.

为解决这个问题,引入一个空间块的概念:

对 n 维空间 $Space_n$ 中的各维定义一个最小跨度,记为 d_{step_i}.一个空间块就是多维空间中的由两个 n 维向量(d_{begin},$span$)确定的区间,d_{begin} 对应 n 维空间中的一个点,称为起始点,d_{begin} 在各维上的投影为该维最小跨度的一个倍数,即 $d_{begin\ i}=n\times d_{step\_i}$.$span$ 为 n 维向量,称为边长表,每一维上的投影同样也是该维最小跨度的一个倍数.一个空间块(d_{begin},$span$)就是一个以 d_{begin} 为起点、向各维的正方向延伸 $span$ 中对应维长度的一个区间.对二维平面而言,一个空间块就是一个矩形.对三维立体空间而言,一个空间块就是一个长方体.

由空间块的定义可知,对 n 维空间 $Space_n$,可以构造 $Space_n$ 的一个划分{(d_0,d_{span}),…,(d_i,d_{spani}),…}.其中(d_i,d_{spani})代表 n 维空间 $Space_n$ 中两两互不相交的空间块.

由空间块的概念和基于密度聚类的思想,引入另一个概念——聚类块.定义变量 ρ,称之为密度.一个聚类块就是 n 维空间 $Space_n$ 上的一个空间块 $block(d_{begin},span)$,$block$ 中包含 n 个数据点,$n\geq\rho\times\prod_{i=1}^{n}span_i$.

在 ACluStream 算法中,子聚类是以聚类块的形式记录的.

1.3 ACluStream完备性和可行性的证明

首先证明完备性:使用聚类块结构能够完整地近似表达任意的空间形状.

当假设数据流的数据具有某种空间特征(由一个特定函数 $f(x)$决定),由 Riemann 积分的几何意义可知,可以把数据流的形状表达问题看成一个多维空间中的图形求体积的问题,即 $f(x)$的 Riemann 积分近似求解问题.同时,根据数值积分的蒙地卡罗方法(Monte Carlo method)得知,可以通过一些相互独立的变量来近似计算欧氏空间中任意函数的定积分,误差范围由样本的数目决定.如对二维欧氏空间,定积分的蒙地卡洛公式为

$$\int_a^b\int_c^d f\left(X^1,X^2\right)\mathrm{d}X^1\mathrm{d}X^2\approx\frac{(b-a)(d-c)}{N}\sum_{i=1}^{N}f\left(X_i^1,X_i^2\right).$$

也就是说,对由函数决定,在某一区间上大量均匀分布、相互独立的变量,其和的均值近似等于定义区间上该函数的积分.所以,当聚类块各维的最小跨度 d_{i_step} 足够小时,可以认为这种基于聚类块的空间结构可以近似地表

达这个 n 维空间中的任意形状.

下面再证明对于任意的聚类形状,如果使用 CluStream 中 micro-cluster 过程计算要消耗的空间复杂度为 $O(S)$,那么最坏情况下,可以使用一个空间复杂度为 $k\times O(S)$的聚类块来表达这个聚类形状.

Micro-Cluster 通过使用多个最大半径为 r 的球形空间子聚类来表达空间聚类.考虑到聚类块的空间意义,显然,最坏情况下,聚类的实际形状就是一个半径为 r 的球形.micro-cluster 只需使用一个子聚类,就能完整地描述该聚类.下面证明,只需使用 k 个聚类块就可以表达这个球形.设各维的步长为 δ_i,那么,最多只要使用 $\prod_{i=1}^{n-1}(r/\delta_i)$ 个块,就可以表达覆盖这个球形的一个正方体.显然,当 n,r 和 δ_i 确定的时候,$\prod_{i=1}^{n-1}(r/\delta_i)$ 是一个常数,不妨将其记为 k.也就是说,只要使用小于 k 个的空间块就可以表达这个球形.所以,即使最坏情况下,算法的空间复杂度也只是 CluStream 的一个线性函数.

1.4 聚类块内对聚类的表达

每一个聚类块使用一个聚类块特征值来表示.聚类块特征值是聚类特征值的一个扩充,它使用一个 7 元组($\overrightarrow{CF^2}$,$\overrightarrow{CF^1}$,$\overrightarrow{CF^0}$,t^2,t,d_{begin},$span$)来标识聚类块在空间中的位置和聚类块中点的分布以及聚类块中点到达的情况.其中,$\overrightarrow{CF^0}$,$\overrightarrow{CF^1}$,$\overrightarrow{CF^2}$ 分别对应于聚类块中点集的数目、算术和以及各维向量的平方和,即统计学上的零阶矩、一阶矩、二阶矩,特别地,将 $\overrightarrow{CF^1}/\overrightarrow{CF^0}$ 称为聚类块的聚类中心;t,t^2 分别表示聚类块中数据点到达时间的和及平方和,d_{begin} 表示该聚类块的起始点,$span$ 代表该聚类块的体积.

1.5 ACluStream算法简述

ACluStream 算法首先积累一定的数据,然后使用任意的聚类算法对它们进行聚类,再把聚类的结果划分成一个个互不相交的聚类块.记录这些聚类块的特征值向量并使用一个 hash 表 H 来记录指向这些向量的指针.然后对每一个新来的数据点 d,如果它属于一个聚类块,那么将其加入该聚类块中,并修改该聚类块的特征值;否则,判定它是否为孤立点.当积累的数据量达到窗口大小时,对内存中的聚类块进行分析,根据实际情况或者结合、或者分解,并将保存在内存中的聚类块特征按 pyramid frame 的结构记录下来,便于后续查询.

2 ACluStream 算法

与 CluStream 相同,ACluStream 也分为两个部分:在线的进程和离线的进程.记录当前数据流特征的在线进程称为 sdmicro-cluster;而离线的响应查询的进程称为 sdmacro-cluster.下面详细介绍两个进程各自的工作.

2.1 sdmicro-cluster过程

sdmicro-cluster 过程使用下面 3 个参数:P_{guess},W,D_{clu}.

P_{guess} 为判断一个点是否为新聚类的概率,取值范围为 0~1 之间的一个实数;W 是数据窗口的大小,取值范围为大于 0 的整数;D_{clu} 为大于 0 的整数,是聚类块的最小密度,代表对 W 个数据,在单位空间 $\prod_{i=1}^{n}d_{step\_i}$ 中至少要包含数据点的数目.

算法描述如下:

对每一个新来的点,先计数,然后进行下面的处理.

如果它属于一个现有的聚类块(这可以很容易地由各个邻近的聚类块的起始点以及聚类边界确定),那么修改该聚类块的聚类块特征值;

否则,这个点或者属于一个新的聚类块,或者是一个独异点.为了节省算法消耗的空间,借鉴数据流中频繁度计算的著名算法 STICKY count[9]的思想,使用一种类似“抛硬币”的方法来猜测是否为该点创建一个新的聚类块:

在 0~1 之间取一个随机数,然后判别它是否比预设的参数 P_{guess} 要小:如果是,那么假设这个点属于一个新的聚类,为它创建一个新的聚类块;否则,认为该点是一个独异点,将它抛弃.

创建一个新的聚类块的过程如下:分配内存中的一个空间,在这个空间中记录一个聚类块的特征.然后,根据聚类块的起点在 hash 表 H 的相应位置记录指向该空间的指针.

当数据流积累到窗口已满的时候,对内存中的聚类块进行下面的运算:

(1) 计算新的密度参数

基于密度的聚类有一个特点,需要预定义聚类的密度.算法使用一个随数据流规模线性递增的密度函数来计算参数,每次数据窗口满了以后,D_{clu} 参数自动增长.

(2) 聚合相邻的、相似的聚类块

为节省系统运行的空间消耗,当数据窗口已满时,聚合相邻的、相似的聚类块.为了确保聚类块足够精确,对聚类块的大小有最大限制,只有当聚类块聚合后仍然满足最大限制时才允许其聚合.

首先判断该聚类块的密度是否还符合定义.如果聚类块的密度超过 D_{clu} 的阈值,那么利用 hash 表 H 计算是否有可以结合的聚类块:如果存在这样的聚类块,且结合后的大小合适,那么将这些聚类块结合并修改聚类块特征值;否则,访问下一个聚类块.

(3) 切割、压缩达不到密度标准的聚类块

如果直接抛弃密度小于 D_{clu} 的聚类块,那么对一些周期性数据流会产生震荡的效果.算法对聚类块中的特征值的时间维度和密度联合考虑,只有当最近一段时间,该聚类块密度的增长低于密度函数的增长 60%时才抛弃.

对密度小于 D_{clu} 又达不到抛弃标准的聚类块,再进一步地判断是否可以对该聚类块进行压缩或分割.由于在每个聚类块中记录了聚类的特征值,通过利用这些特征值,可以在每一维精确地计算方差.如果在某一维上的方差超过某一阈值 S,且该维上的聚类中心 $\overrightarrow{CF_i^1}$ 与聚类块中心 $d_{begin\_i}+\frac{1}{2}span\_i$ 的偏离度较大,那么对该维进行切割:把该聚类块分成两块,保留聚类中心点所在的一块,适当修改聚类特征值;否则,若聚类中心与聚类块中心紧密结合在一起,而且在该维上数据的方差不大,那么以一个合理的尺度在该维上收缩聚类块的大小,修改聚类特征值.

(4) 储存内存中的聚类块特征

CluStream 每隔一段时间就储存内存中记录的所有子聚类及其特征,称之为 snapshot,并按 pyramid time frame 的策略组织这些 snapshot.Pyramid time frame 是一种按时间来组织数据流的初步描述的策略,它保证了对一个用户定义的时间窗 h,在当前时间的 $2h$ 窗口内至少存在一个 snapshot.ACluStream 同样采取 pyramid frame 结构储存数据,关于 pyramid frame 的思想和具体做法可见文献[4].

2.2 sdmacro-cluster过程

sdmicro-cluster 得到的结果以 pyramid frame 结构储存在磁盘上.然后使用 sdmacro-cluster(shape density macro-clusterstream)过程响应查询.sdmacro-cluster 的作用就是比较两个不同时间的结果.CluStream 使用对子聚类编号的方法来实现子聚类的标认,并据此判断子聚类的变化.但在 AcluStream 算法中,由于聚类块有严格的空间意义,因此不需要对其进行额外的编号.

由于 sdmacro-cluster 的相对独立,可采取任意一种支持加权聚类的算法来进行.本文对此并不作详细讨论.

2.3 算 法

sdmicro-cluster(P_{guess},W,D_{clu},$Datastream$)

{先积累一段时间的数据流于滑动窗口,然后用基于密度的方法对数据流聚类,并将聚类结果划分为互不相交的聚类块.

```
For (不到数据流末尾){
    For(i=0;i<w;i++){
        读数据流中的一个点;
```

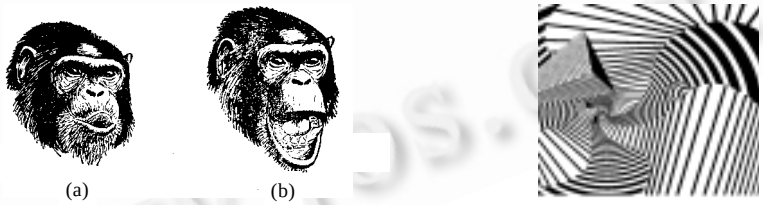
根据点的位置以及数据块在各维上的最大长度,利用 hash 表搜索是否存在一个包含这个点的聚类块;
if (exist){
修改此聚类块特征;
}else{
if (P_{guess}<*getrandom*())为该点创建一个聚类块,并将其加入 hash 表;
}}
按累计数据的规模重新计算密度函数 D_{clu};
按 hash 表中顺序扫描内存中的聚类块{
if (块内密度≤D_{clu}){if (块内密度大于可抛弃的密度){
切割或收缩该块;
}else{
释放该聚类块的空间;};
}else{if (exist(可聚合的聚类块)){聚合}
}}
按 pyramid frame 的结构储存数据;
}}

3 算法性能

为了比较 CluStream,ACluStream 的聚类效果,我们分别按照 2 幅 800×600 的位图构造数据流,并根据一定比例随机生成噪声点以作为数据源.然后,使用相同大小、数目的子聚类来对数据流进行聚类.当在线程序处理了 1 000 000 个数据点后,分别比较两种算法聚类形状、聚类质量以及算法的运行速度.

3.1 数据流的形状比较

使用以下两幅图来生成数据流.实验数据流按图 2 中的两幅图形的分布随机生成,并按 10%的比例生成噪声点.



(a) (b)

(a) Picture 1 (b) Picture 2
(a) 数据源 1 (b) 数据源 2

Fig.2 The pictures for generating the data
图 2 实验数据源

产生数据源 1 的图是两个表情不同的猩猩头像,这两个头像分别标注为(a)和(b).可以通过它来比较两种算法在处理现实图形数据时的区别.产生数据源 2 的图是一幅由多种几何图形组成的图片.通过它来比较两种算法在处理几何图形数据时的区别.

由于 CluStream 算法的聚类效果对子聚类数目和子聚类的面积十分敏感,实验中分别采取了多种不同的子聚类数目及半径.多次实验比较的结果证明,ACluStream 算法的聚类结果要优于 CluStream 的聚类结果.由于篇幅所限,这里只给出实验中子聚类数最多、半径最小时的实验结果.这次实验中,取子聚类的最大面积为 18,数据源 1 共使用了 9 293 个聚类块,数据源 2 共使用 11 895 个聚类块.图 3、图 4 展示了两种算法的实验结果,其中图的左半部分是 ACluStream 算法得到的结果,右半部分是 CluStream 算法得到的结果.

Fig.3 Comparison of accuracy in picture 1's data

图 3 数据源 1 的实验结果

由图 3 的实验结果可以直观地看到:在对真实图像表达能力方面,尤其在对一些细节的描述上,ACluStream 优于 CluStream.观察猩猩伸舌头的动作以及标注(a),(b),它们在 ACluStream 的结果中基本还原,但在 CluStream 结果中就有较大的失真.

Fig.4 Comparison of accuracy in picture 2's data

图 4 数据源 2 实验结果

由图 4 的实验结果可以直观地看到:在几何图像表达能力方面,尤其在处理一些间距比较密的线条时,ACluStream 也明显优于 CluStream.下面,使用量化的数据来比较两个算法得到的结果.

对两幅图 G_1,G_2 的相似度定义为:在 G_1,G_2 取值相同的坐标所占图形面积的比例.图 5 是所作实验的一些比较结果.横坐标表示所使用的聚类的不同精度,纵坐标表示聚类结果的相似度.

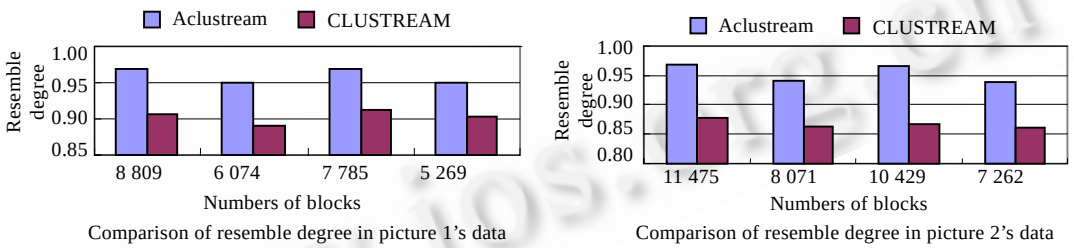
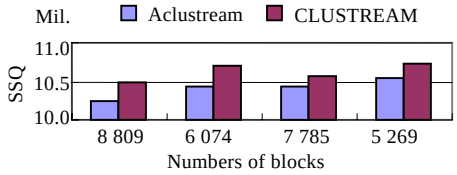


Fig.5 Comparison of resemble degree in experimental data

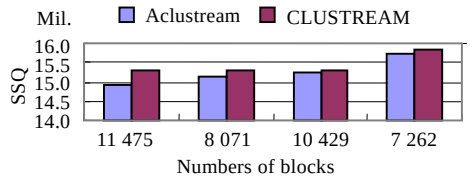
图 5 ACluStream 与 CluStream 相似度比较

3.2 聚类质量比较

SSQ(sum of square distance)是一种比较 k-划分聚类质量的方法,它通过计算所有点到各自的聚类中心的距离来衡量算法所给出的 k-划分的质量.SSQ 值越小,说明算法聚类质量越好.我们使用 k-means 方法作为 ACluStream 算法和 CluStream 算法的离线聚类过程来比较两个程序的初始聚类质量,然后比较两个程序算出的 SSQ 值.有趣的是,虽然 ACluStream 采用一个基于密度的方法来对数据进行初步聚类,但使用它却得到更好的结果.继续上面实验的比较,如图 6 所示.



Comparison of SSQ in picture 1's data



Comparison of SSQ in picture 2's data

Fig.6 Comparison of SSQ in experimental data

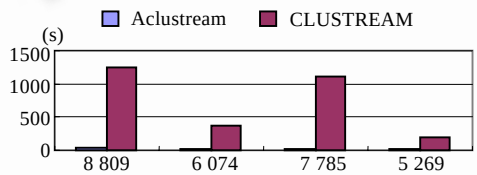
图 6 AClusStream 与 CluStream 的 SSQ 值比较

3.3 算法速度比较

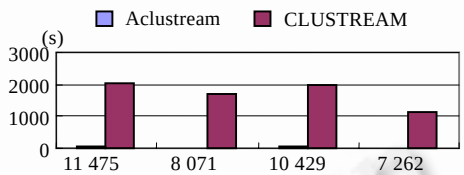
最后比较算法的运行速度.从本文的第 2 节对 micro-cluster 过程的分析可知,对程序的效率影响主要有两方面的原因:(1) 对用于初步记录聚类信息的子聚类数据结构而言,每当一个新点到达时,其聚类中心点都会改变,需要修改索引;(2) 由于算法的限制,会频繁地出现子聚类聚合和增删操作,从而降低了程序的效率.

而 AClusStream 算法采用严格的空间定义,所以很容易地就建立了 HASH 索引,这个索引相对稳定,只有当数据窗口已满,有聚类块需要切割/合并时才需要对所涉及的聚类块进行修改,极大地提高了程序的效率.所以两个程序在处理数据速度上有很大的差别.

进行算法速度测试的具体运行环境是:pIII 700 cpu,256m ddr ram 操作系统为 win2k server,下面是程序运行的结果(如图 7 所示).



Comparison of time consume in picture 1's data



Comparison of time consume in picture 2's data

Fig.7 Comparison of time consume in experimental data

图 7 AClusStream 与 CluStream 运行速度比较

4 结论与进一步研究的方向

实验结果显示,当 AClusStream 使用聚类块与 CluStream 子聚类数目相同、最大面积相等时,无论是时间效率还是准确性,AClusStream 算法都比 CluStream 要好.

不过,无论是 CluStream 还是 AClusStream 算法,它们都是基于欧氏空间的算法.而且仔细分析算法的流程可以发现,它们都利用了欧氏空间距离的一些特性,不太适合解决非欧氏空间的问题.如何对非欧氏空间的数据流(文本流、多媒体流)进行高效的聚类分析,将是我们下一步工作的方向.

References:

[1] Golab L, Özsu MT. Issues in data stream management. SIGMOD Record, 2003,32(2):5−14.

[2] Babcock B, Babu S, Datar M, Motwani R, Widom J. Models and issues in data stream systems. In: Proc. of the 21st ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems. 2002. 1−16.

[3] Barbará D. Requirements for clustering data streams. ACM SIGKDD Explorations Newsletter, 2003,3(2):23−27.

[4] Aggarwal C, Han J, Wang J, Yu PS. A framework for clustering evolving data streams. In: VLDB 2003. 2003. 81−92.

[5] Guha S, Mishra N, Motwani R, O'Callaghan L. Clustering data streams. In: FOCS 2000. 2000. 359−366.

[6] O'Callaghan L, Mishra N, Meyerson A, Guha S. Streaming-Data algorithms for high-quality clustering. In: ICDE Conf. 2002. 685−704.

[7] Zhang T, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases. In: SIGMOD'96. 1996. 103−114.

[8] Han J, Kamber M. Data Mining-Concepts and Techniques. Beijing: Higher Education Press, Morgan Kaufmann Publishers, 2001.

[9] Manku GS, Motwani R. Approximate frequency counts over data streams. In: VLDB 2002. 2002. 346−357.

朱蔚恒(1976−),男,广东广州人,博士生,主要研究领域为数据流挖掘,数据流应用.

谢益煌(1981−),男,硕士生,主要研究领域为数据挖掘.

印鉴(1968−),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据挖掘,机器学习.