一种新的数据流分形聚类算法

罗义钦[1,2],倪志伟[1,2],杨葛钟啸[1,2]
LUO Yi-qin[1,2],NI Zhi-wei[1,2],YANGGE Zhong-xiao[1,2]

1.合肥工业大学 管理学院,合肥 230009
2.合肥工业大学 过程优化与智能决策教育部重点实验室,合肥 230009
1.School of Management,Hefei University of Technology,Hefei 230009,China
2.Key Laboratory of Process Optimization and Intelligent Decision-making,Ministry of Education,Hefei University of Technology,Heifei 230009,China

LUO Yi-qin,NI Zhi-wei,YANGGE Zhong-xiao.New fractal clustering algorithm on data stream.Computer Engineering and Applications,2010,46(6):136-138.

Abstract: This paper presents an algorithm which is based on fractal to cluster data stream and uses the change of fractal dimension to measure the self-similarity between data and clusters.With noisy condition,the algorithm can discover arbitrary shape clusters that reflect the natural group status of data stream.The experiments show the good performance and effectivity of FClustream.

Key words: data stream;fractal;fractal dimension;clustering

摘 要:提出了基于分形的数据流聚类算法,利用分形维数的变化程度来度量数据点与聚类的自相似程度,在噪音干扰下能发现反映数据流自然聚集状态的任意形状的聚类。实验证明,FClustream 算法是一种高效的数据流聚类算法。

关键词:数据流;分形;分形维数;聚类
DOI:10.3778/j.issn.1002-8331.2010.06.039 **文章编号:**1002-8331(2010)06-0136-03 **文献标识码:**A **中图分类号:**TP301.6

聚类是将客观对象的集合根据对象之间的某种相似性分组组成多个集合的过程,其中同一集合的对象之间必须彼此相似,属于不同集合的对象必须彼此相异。聚类的本质在于基于某种度量的相似性。空间距离的接近、空间形状的相近、密度的相近、整体分布的相似性及以上因素的综合都已经被用于聚类算法。分形理论的核心思想是局部与整体之间的自相似性。从这个意义上说,分形维数可以作为聚类的一种度量。2000 年,Daniel Barbara 首先提出了基于分形的聚类方法[1]。

数据流给数据挖掘算法提出了新的挑战。针对数据流持续到达,且速度快、规模大等特点,数据流聚类算法不仅要求能进行单遍扫描,而且要求利用有限的内存资源和存储资源进行聚类。提出的基于分形的数据流聚类算法满足以上要求,不仅能够发现任意形状的聚类,而且能够发现传统聚类无法发现的数据流隐含的模式。

1 相关研究

数据流聚类问题一直是国内外研究者关注的研究热点,已经提出了多种数据流聚类算法。

文[2]提出 LOCALSEARCH 数据流聚类算法。算法基于分治的思想使用一个不断的迭代过程实现利用有限空间对数据流进行 K-means 聚类。文[3]发展了 Guha 的 LOCALSEARCH 的思想,提出了 STREAM 算法,并利用 SSQ 证明在多种情况下,STREAM 算法的聚类效果要比 BIRCH 好。但这两种算法都只能提供对当前数据流的一种描述,而不能反映数据流的变化情况。

文[4]提出了一个优秀的解决数据流聚类框架——CluStream 算法。它使用了两个过程来处理数据流聚类问题:首先,使用一个在线的微观聚类过程对数据流初步聚类,并按照一定的时间跨度将聚类结果按金字塔时间结构存储,以便进行离线分析。同时,使用一个离线的宏观聚类过程,根据用户的要求对宏观聚类过程的聚类结果进行再分析。但它采用距离作为相似度度量,聚类结果通常是球形的,不能支持任意形状的聚类。

文[5]提出了 DenStream 算法。它能够对有噪声的动态进化数据流进行任意形状的聚类。它的缺点是由于采用了全局一致的绝对密度作为参数,造成了对参数十分敏感,同时不能反映用户指定时间跨度内的数据流的变化情况。

AClustream 聚类算法是文[6]提出的基于密度与空间的数据流聚类算法。它通过引入有严格空间的意义聚类块,尽量保

基金项目:国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.2007AA04Z116);国家自然科学基金重点项目(the Key National Natural Science Foundation of China under Grant No.70871033)。

作者简介:罗义钦(1983-),男,硕士生,主要研究方向:数据流挖掘、人工智能;倪志伟(1963-),男,教授,博士生导师,主要研究方向:人工智能、机器学习;杨葛钟啸(1984-),男,硕士生,主要研究方向:分形理论及应用。

收稿日期:2008-09-10 **修回日期:**2008-11-21

留数据的空间特性，有效克服 CluStream 算法不能对任意形状聚类的缺陷。它的缺点是在处理不属于已有聚类块的新数据点时误差较大,同时不能区分密度等级不同的簇。

文[7]提出的基于网格的 GClustream 算法是借鉴了 CluStream 算法的两阶段聚类的框架和 pyramid time frame 的存储结构,采用相对密度作为聚类参数,通过对数据流进行网格处理,能在噪声条件下发现任意形状的聚类。它的缺点是对于处理高维空间中的分布稀疏、噪声难以区分的数据流的聚类结果不好,需要进行降维处理。

提出的基于分形的数据流聚类算法 FClustream(Fractal Based Data Stream Clustering Algorithm),采用 CluStream 算法的数据流聚类框架[4],分为两个过程。在线聚类过程采用分形维数作为聚类度量,通过判断数据流中数据点对各个聚类的分形维数的影响程度,对数据流进行聚类和发现噪声点;同时,用户可以通过离线分析过程,从存储在磁盘上的数据流快照得到与设定参数对应的挖掘结果和给定时间段内到达数据点的聚类情况,从而分析数据流聚类情况和变化过程。FClustream 算法满足数据流聚类条件,提高了算法的处理速度,并能在噪声条件下发现任意形状的类,能够发现数据流中隐藏的模式。

2 分形维数

2.1 分形的数学基础

大量研究证明:分形是自然界中普遍存在的现象,如股票交易数据、人类的生理现象、网络的通信行为等都是自相似的。

对于 n 维空间中的数据,将它嵌入到每一个单元格(Cell)边长为 $r(r[r_{\min},r_{\max}])$的 n 维格子中,$[r_{\min},r_{\max}]$是数据集具有分形特性的边长度量的变化范围。这样,可以计算出落入第 i 个单元格中数据点的数目,记作 C_i^q。

定义 1(分形维数) 对于一个在区间$[r_{\min},r_{\max}]$(无标度区)内呈现统计自相似特征的数据集,其分形维数 D_q 定义如下:

$$D_q=\frac{1}{q-1}\frac{\partial\log\Sigma_i C_i^q}{\partial\log r},r\in[r_{\min},r_{\max}] \tag{1}$$

由于数据集的数据点是有限的,因此其在一定区间尺度内具有统计自相似性,即 $r\in[r_{\min},r_{\max}]$。当 $q=0$ 时为豪斯道夫维数,$q\to1$ 时为信息维,而 $q=2$ 时为关联维。研究表明[6],信息维和关联维对于数据挖掘特别有用。信息维的变化说明了数据集的变化趋势,关联维的变化说明了数据集中的点分布的变化情况。

2.2 分形维数在数据流中的计算方法

分形维数在数据流的计算必须符合数据流持续到达,且速度快、规模大等特点,要求能进行单遍扫描,而且要求利用有限的内存资源和存储资源进行计算。文[8]提出的 SID 方法符合上述要求。该方法提出一种利用 SID 表模块的存储结构建立一棵分形树,用来记录对于不同单元格边长 r_j($r_j=r_{j-1}/2$),落入每个单元格中的数据点数。SID 表模块包含一个用来判断这个单元格是否属于上一层的某个单元格的区分器 $identifier[b_1b_2\cdots b_E]$,一个用来记录落入每个单元格中的数据点数的计数器数组 $C[k]$,一个该单元格指向下一层的第一个节点的指针 P_c 和指向同一层的兄弟节点的指针 P_b。该方法首先计算出数据点的每一个属性的最大值 rh_i 和最小值 rl_i,得到 $r_0=\max(rh_i-rl_i),i=1,2,\cdots,E$,将每维的取值范围 2^j 等分($j=0,1,\cdots,level$),其中,$level$ 为分形树的层数。除根节点之外,每个节点由 2^E 个单元格的信息组成。因此在理论上，整个数据流的数据集合被分成 2^{Ej} 个单元格,但在实际中只记录计数数组 $C[k]>0$ 的单元格。这样就得到了一棵树高为 $level$ 的分形树。令 $C_{r_j,i}$ 为数据集中的数据点落入第 j 层的边长为 r_j 的第 i 个单元格中的数据点数。对每一层中的单元格进行统计可以得到一系列点$\{d_0,d_1,\cdots,d_j,\cdots,d_{level}\}$,其中 $d_j=<\log(\sum_{i=0}^{i<2^E}C_{r_j,i}^2),\log(r_j)>,j=0,1,\cdots,level$。对$\{d_0,d_1,\cdots,d_j,\cdots,d_{level}\}$进行拟合,则近似为直线部分的斜率可以作为数据集的关联维的近似值。

3 基于分形的数据流聚类算法

3.1 数据结构及概念

设 $A=\{A_1,A_2,\cdots,A_E\}$是有界域的集合,$S=A_1\times A_2\times\cdots\times A_E$是一个 E 维数据空间,其中 $A_1,A_2,\cdots,A_E$ 表示 S 的 E 个属性域或 E 个维。$X=\{x_1,x_2,\cdots,x_N\}$表示 S 上的 N 个点的集合,其中,$x_i=\{x_{i_1},x_{i_2},\cdots,x_{i_E}\}$表示一个数据点,$x_{i_j}\in A_j$ 表示数据点 x_i 的第 j 维的值。

为了便于计算数据流的分形维数，用 SID 算法构建分形树,方法如上一章所述。

定义 2(分形影响度) 数据点 e 加入数据集 X 得到新数据集 X',$F_q(X)$和 $F_q(X')$分别为数据集 X 和新数据集 X'的分形维数,数据点 e 对数据集 X 的分形影响度 FI 定义如下:

$$FI=|F_q(X')-F_q(X)| \tag{2}$$

根据分形树的结构,只要保存各层的单元格及其相应的数据点数目信息,容易得到数据集的分形维数,从而得到数据点的分形影响度。

对于数据点来说，它对不同数据集的分形影响度是不同的。分形影响度的大小直接地反映了数据点(部分)和数据集(整体)之间的自相似性。分形影响度越小,自相似性就越大。分形影响度越大,自相似性就越小。

3.2 FClustream 算法描述

分形维数是反映数据分布复杂程度的重要特征参数,随着数据结构形态的不同而变化,在一定程度上反映了数据集内部结构的复杂性。同一数据集的分形维数不会因为空间维数和位置不同而发生显著的变化。属于该数据集的数据点对该数据集的分形影响度是非常小的。基于这个思想,提出了 FClustream 算法对数据流中的数据点进行聚类。

FClustream 算法主要分为初始聚类、在线聚类和离线进程三个过程,其中初始聚类过程可以使用当前的任何一种聚类算法。FClustream 算法的具体描述如下。

3.2.1 初始化

先积累一段时间的数据流,然后使用当前的任何一种聚类算法进行聚类。设共聚成个 k 类,记为$\{C_1,C_2,\cdots,C_k\}$,分别对初始化的类计算其分形维数,并建立分形树$\{FTree_1,FTree_2,\cdots,FTree_k\}$,第 i 个类的分形维记为 $F_d(C_i)$。

3.2.2 在线聚类

步骤 1 把数据流的数据点存储在大小为 W 时间跨度为 Δt_c 的滑动时间窗口 $W(t_i)$中。$W(t_i)$按时间先后划分成 η 个等宽的子窗口,称为基本窗口。基本窗口的时间跨度为 $\Delta t_c/\eta$。

步骤 2 对于每一个点 $e \in W(t_i)$,将 e 加入每一个类中,得到 $C_i'=C_i \cup \{e\}$,并对每一棵分形树 $FTree_i$ 进行更新。分形树的每一个单元格只保持最近的 λ 个基本窗口(时间跨度根据实际需求定义)的 $C[k](m)$,$k=1,2,\cdots,\lambda$,其中 m 表示单元格所在的层数,因此计数器数组 $C[k]$ 有 λ 个数组成员,记录最近的时间跨度为 $\lambda \times \Delta t_c/\eta$ 的数据流的计数情况,每一个数组成员对应一个小的时间跨度落入单元格的数据点数。当 λ 个基本窗口的数据点计算结束,开始计算下一个基本窗口的数据点时,应丢弃与最早的基本窗口相对应的计数器。该单元格的根节点及其左右子树中与数据点落入位置对应的单元格也要进行相同的操作。

步骤 3 计算 C_i' 的分形维数 $F_d(C_i')$。

步骤 4 在 k 个类中找到 e 加入后使得原来类的维数变化最小的一个类,即分形影响度最小的一个类,记为 $\hat{i}$,$\hat{i}=\min_i FI=\min_i |F_d(C_i')-F_d(C_i)|$。

步骤 5 如果 $\min|F_d(C_i')-F_d(C_i)|$ 小于预定的一个阈值 ε,则认为 $e \in C_i$,对 $i \neq \hat{i}$ 的分形树 $FTree_i$ 进行还原,还原成未添加 e 之前的状态,否则,认为该点是离群点,删除离群点,还原每一个分形树 $FTree_i$,还原成未添加 e 之前的状态。

3.2.3 离线分析进程

如果完成 $W(t_i)$ 中的所有数据点的聚类,就将 $\{FTree_1, FTree_2, \cdots, FTree_k\}$ 存储到磁盘,并按照 CluStream 算法提出的金字塔时间结构的策略组织这些结果。

用户可以根据自己设定的参数和查询时间跨度,从存储在磁盘上的数据流快照得到与设定参数对应的挖掘结果和给定时间段内到达数据点的聚类情况,从而分析数据流聚类情况和变化过程。

4 实验与分析

4.1 实验结果分析

算法实验所用的所有代码,均采用 Microsoft Visual C# 实现,实验环境是一台 P4 2.4 GHz CPU 的联想电脑,具有 512 MB 主存。采用了 UCI 的 KDD CUP 1999 数据集,该数据集共有数据点 4 898 431 个。KDD CUP 1999 数据集共有 23 类,每一数据有 41 个属性。其中 normal、neptune 和 smurf 这三类最多,占总数据集的 98%。取出这 3 类作为实验数据,每一个类取 19 000 条和 300 个离群点。首先从每类中取出 1 000 条数据进行初始化,然后模拟数据流环境,将剩下的 54 300 条数据分成 3 份,分别在 t_0,t_1,t_2($t_0<t_1<t_2$)三个时刻从 54 300 条数据不重复地随机抽取 18 100 条的数据,以每秒 100 条的速率让 FClustream 算法计算。把判断是否为离群点的阈值 ε 设置为 0.01,分形树高设为 $level$=10,滑动窗口的划分为 η=10 个基本窗口,滑动窗口时间跨度为 Δt_c=181 s,计数器数组的参数 λ=10。最终的聚类结果如表 1 所示。

表 1 FClustream 聚类情况表

类名	实际记录数	聚类后记录数	错分点数	错分率/(%)	正确率/(%)
normal	19 000	21 185	0	0	100.00
neptune	19 000	18 699	301	1.58	98.42
smurf	19 000	17 116	1 884	9.92	90.08
outlier	300	300	0	0	100.00

表 1 所示,算法共错分数据点 2 185 个,约占总点数的 3.8%,那么聚类准确率约为 96.2%,证明了 FClustream 算法的有效性。

分别在 t_1,t_2 时刻提出聚类请求,计算聚类准确率和空间开销,并与 DGStream 算法[9]进行比较(如图 1)。可以看到 FClustream 算法的聚类效果略高于 DGStream 算法,而且随着时间的推移聚类效果还会更好一点。因为分形维数的盒子计数法的准确度最优的情况是用最少的盒子包含所有数据点。随着时间的推移,由于数据量越大越能体现出自相似性,数据点的积累有利于反映数据流变化的真实情况,越来越能体现数据流的聚集特点,同时也有利于提高盒子计数法的计算准确度。

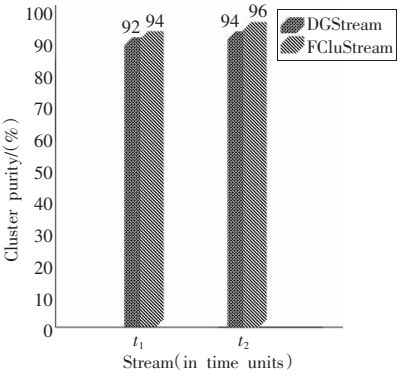

图 1 FClustream 算法与 DGStream 算法聚类有效性比较

4.2 算法效率分析

4.2.1 时间复杂度分析

FCluStream 算法的时间复杂度主要受计算聚类结果的分形维数的影响,因此着重分析计算聚类结果的分形维数的时间复杂度。

计算分形维数需要得到分形树不同层次的单元格所包含点的统计信息,因而需要对不同层次的单元格进行扫描工作。因为算法对每层的单位格的半径进行了有规律的划分,同时记录下了每一个单元格的兄弟指针和孩子指针,所以对每层分形树的扫描的时间复杂度是 $O(N\log N)$,其中 N 是聚类中数据点的总数。要对整个分形树进行扫描工作,那么总的时间复杂度是 $O(level \cdot N\log N)$,其中 $level$ 为分形树的层数。

4.2.2 空间复杂度分析

FCluStream 算法的空间复杂度主要来自计算聚类结果的分形维数的所存储的分形树所需的存储空间。分形树中的每层单元格的总数几乎就等于聚类中数据点的总数 N。存储一个聚类对应的分形树所需的主内存存储空间是 $O(level \cdot N)$,其中 $level$ 为分形树的层数。因为 FCluStream 算法只更新数据点对应的单元格中的计数器,不存储数据点的具体信息,因此,算法所需的存储空间独立于数据点的维数。算法是对每个聚类建立一棵分形树,那么整个算法的空间复杂度是 $O(level \cdot N \cdot cluster)$,其中 $cluster$ 为聚类的个数。

5 小结

FCluStream 算法基于分形技术,结合数据流的特点,通过计算数据点对聚类的分形维数的扰动进行聚类,并提出在线聚类过程和离线分析过程,从而支持用户可以通过这两种方式分析数据流的聚类结果。通过实验分析发现,算法表现出很好的时间特性,聚类结果会随着时间的推移表现得更好。在实际数

(下转 143 页)