

基于关联函数的数据流聚类算法

潘丽娜*, 王治和, 党辉

(西北师范大学 计算机科学与工程学院, 兰州 730070)

(*通信作者电子邮箱 panln84@163.com)

摘要:传统数据流聚类算法大多基于距离或密度,聚类质量和处理效率都不高。针对以上问题,提出了一种基于关联函数的数据流聚类算法。首先,将数据点以物元的形式模型化,建立解决问题所需要的关联函数;其次,计算关联函数的值,以此值的大小来判断数据点属于某簇的程度;然后,将所提方法运用到数据流聚类的在线-离线框架中;最后,采用真实数据集 KDD-CUP99 和随机生成的人工数据集进行算法的测试。实验结果表明,所提方法的聚类纯度在 92% 以上,每秒能处理约 6300 条记录,与传统算法相比,处理效率有了较大的提高,在维度和簇数目方面的可扩展性较强,适用于处理大规模的动态数据集。

关键词:数据流;聚类;物元;关联函数;经典域;节域

中图分类号: TP311.5 **文献标志码:** A

Data stream clustering algorithm based on dependent function

PAN Lina*, WANG Zhihe, DANG Hui

(School of Computer Science and Engineering, Northwest Normal University, Lanzhou Gansu 730070, China)

Abstract: The traditional data stream clustering algorithms are mostly based on distance or density, so their clustering quality and processing efficiency are weak. To address the above problems, this paper proposed a data stream clustering algorithm based on dependent function. Firstly, the data points were modeled in the form of matter-element and dependent function was established to solve the problem. Secondly, the value of the dependent function was calculated. According to this value, the degree that data point belongs to a certain cluster was judged. Then, the proposed method was applied to online-offline framework of the data stream clustering. Finally, the proposed algorithm was tested by using the real data set KDD-CUP99 and randomly generated artificial data sets. The experimental results show that clustering purity of the proposed method is over 92%, and it can deal with about 6300 records per second. Compared with the traditional algorithm, the processing efficiency of the algorithm is greatly improved. In the aspects of dimension and the number of cluster, the algorithm shows stronger scalability, and it is suitable for processing large dynamic data set.

Key words: data stream; clustering; matter-element; dependent function; classical domain; joint domain

0 引言

现代计算机网络和传感器网络技术的快速发展引发了许多具有广阔发展前景的数据流应用技术——从环境和天文监测、电力供应网、金融股票交易到电子商务记录等。这些应用中,海量数据以实时方式快速到达,相对于传统静态数据存在形式,数据流已成为一种新兴且日益主流的数据存在方式。作为一种新的数据形态,数据流对数据挖掘提出了诸多挑战。迄今为止,学者们已给出了大量处理数据流的挖掘算法,典型算法有:2000 年 Guha 等提出了针对数据流聚类的算法 Local Search^[1],该算法基于分治的思想,使用迭代过程对数据流进行聚类。O'Callaghan 等于 2002 年提出 Stream 算法^[2],该算法采用批处理方式对数据进行分级聚类。2003 年 Aggarwal 等提出了一种解决数据流聚类演化问题的框架——CluStream 算法^[3],该框架由在线和离线两阶段构成,在线阶段生成数据流的统计信息微簇,离线阶段利用存储的微簇进行聚类。在此基础上,Aggarwal 等于 2004 年又提出了 HPStream 算法^[4],采用投影技术解决了数据流的高维问题,并使用衰减因子不断衰减历史数据。2006 年 Cao 等提出的 DenStream 算法^[5]扩展了传

统聚类算法中基于密度的方法 DBSCAN,使得 DenStream 算法可以处理任意形状的数据流。朱蔚恒等^[6]针对 CluStream 算法对非球形的聚类效果不好和对周期性数据的聚类变化反映不完整等问题提出了一种采用空间分割、组合以及按密度聚类的算法 ACluStream。2010 年张晨等^[7]给出了一种不确定数据流算法,重新定义了不确定簇的聚类特征与质量标准。王述云等^[8]利用距离和熵来定义对象间的相似度。

通过对已有算法的总结和分析可以发现:目前的数据流聚类算法大多数都沿用了 CluStream 框架,此框架是数据流聚类研究的一个重大突破,因此本文选取 CluStream 算法作为比较算法。尽管 CluStream 算法有效地解决了数据流动态变化的问题,但是由于在聚类时采用 K-means 算法,所以不可避免地将该算法固有的缺陷带入到 CluStream 算法中。因为 K-means 算法需要用户指定生成的簇数目和采用平方误差准则,这就造成了 K-means 算法不能有效发现数据集中的自然簇数目和对离群点敏感等问题,从而也大大降低了 CluStream 算法处理数据流的聚类质量。针对以上提及的 CluStream 算法中的不足,本文提出了一种基于关联函数^[9-12]的数据流算法——EXCluStream 算法,在此算法中将保留在线-离线两阶

收稿日期:2012-07-26;修回日期:2012-08-27。

基金项目:甘肃省科技支撑计划项目(090GKCA075);2012 年度教育部人文社会科学研究项目(12YJCZH282)。

作者简介:潘丽娜(1984-),女,湖北武汉人,硕士研究生,主要研究方向:数据挖掘;王治和(1965-),男,甘肃武威人,教授,主要研究方向:数据库技术、数据挖掘;党辉(1988-),女,甘肃刘家峡人,硕士研究生,主要研究方向:数据挖掘。

段框架。EXCluStream 算法利用可拓学^[13-14]中关联函数值的大小来代替传统聚类中的距离和相似系数,并以此度量聚类对象之间的接近度和相似程度。算法开始时,使用初始微簇算法进行聚类的初始化,然后将初始微簇算法用作在线更新微簇以及离线宏聚类的基础算法,这样的处理能够使 EXCluStream 算法具有很好的聚类精度和较低的时间复杂度。

1 预备知识

假设存在一张病人信息登记表,以此表 1 来说明以下的定义 1~5(表中的病人序号属性不作为聚类过程中的有用特征属性)。

表 1 病人信息登记表

病人序号	年龄	身高/cm	体重/kg	入院天数/d
1	18	164	55	3
2	22	170	65	12
3	26	165	60	10
4	20	160	45	6
5	30	178	75	15

定义 1 物元。以物 O 为对象, c 为特征, O 关于 c 的量值 v 构成的有序三元组 $M = (O, c, v)$ 作为描述物的基本元,称为一维物元。物 O 的 d 个特征 c_1, c_2, \dots, c_d 及 O 关于 $c_j (j = 1, 2, \dots, d)$ 对应的量值 $v_j (j = 1, 2, \dots, d)$ 所构成的阵列

$$M = \begin{bmatrix} O, & c_1, & v_1 \\ & c_2, & v_2 \\ & \vdots & \vdots \\ & c_d, & v_d \end{bmatrix} = (O, C, V)$$

称为 d 维物元,其中:

$$C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_d \end{bmatrix}, V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix}$$

数据点与物元的概念是等同的,物元的 d 个特征即为数据点的 d 维属性,量值即为数据点在第 j 维属性上的具体取值。以表 1 中病人序号为 1 的记录作为说明,其物元形式为:

$$M = \begin{bmatrix} \text{病人 1} & \text{年龄} & 18 \\ & \text{身高} & 164 \\ & \text{体重} & 55 \\ & \text{入院天数} & 3 \end{bmatrix}$$

其他病人的物元表示类似。

定义 2 经典域。根据可拓学理论, $N_i (i = 1, 2, \dots, m)$ 是全体物元对象集合的 m 个子集,由物元模型的基本概念构建经典域如下:

$$R_i = (N_i, C, Y_i) = \begin{bmatrix} N_i, & c_1, & Y_{i1} \\ & c_2, & Y_{i2} \\ & \vdots & \vdots \\ & c_d, & Y_{id} \end{bmatrix} = \begin{bmatrix} N_i, & c_1, & \langle a_{i1}, b_{i1} \rangle \\ & c_2, & \langle a_{i2}, b_{i2} \rangle \\ & \vdots & \vdots \\ & c_d, & \langle a_{id}, b_{id} \rangle \end{bmatrix}$$

其中: N_i 对应将全体物元对象划分后的第 i 簇 ($i \leq m$); $c_j (j = 1, 2, \dots, d)$ 为簇 N_i 中包含的物元对象的 d 个不同的属性; Y_{ij} 为 N_i 关于相应特征 c_j 的量值范围,即为经典域。记 $Y_{ij} = \langle a_{ij}, b_{ij} \rangle (i = 1, \dots, m; j = 1, \dots, d)$, 则 R_i 为第 i 簇经典域的物元模型。假设已划分序号为 1, 4 的病人在簇 1 中,序号为 2, 3, 5 的病人在簇 2 中,则构建簇 1、簇 2 的经典域如下:

$$R_1 = \begin{bmatrix} \text{簇 1} & \text{年龄} & \langle 18, 20 \rangle \\ & \text{身高} & \langle 160, 164 \rangle \\ & \text{体重} & \langle 45, 55 \rangle \\ & \text{入院天数} & \langle 3, 6 \rangle \end{bmatrix}$$

$$R_2 = \begin{bmatrix} \text{簇 2} & \text{年龄} & \langle 22, 30 \rangle \\ & \text{身高} & \langle 165, 178 \rangle \\ & \text{体重} & \langle 60, 75 \rangle \\ & \text{入院天数} & \langle 10, 15 \rangle \end{bmatrix}$$

定义 3 节域。为其相应经典域的并集,即定义 $c_j (j = 1,$

$2, \dots, d)$ 的节域 $Y_{pj} = \bigcup_{i=1}^m \langle a_{ij}, b_{ij} \rangle$, 设 $\langle a_{pj}, b_{pj} \rangle = \bigcup_{i=1}^m \langle a_{ij}, b_{ij} \rangle$, 则可将各项特征的节域以物元模型 R_p 表示为:

$$R_p = (N_p, C, Y_p) = \begin{bmatrix} N_p, & c_1, & Y_{p1} \\ & c_2, & Y_{p2} \\ & \vdots & \vdots \\ & c_d, & Y_{pd} \end{bmatrix} = \begin{bmatrix} N_p, & c_1, & \langle a_{p1}, b_{p1} \rangle \\ & c_2, & \langle a_{p2}, b_{p2} \rangle \\ & \vdots & \vdots \\ & c_d, & \langle a_{pd}, b_{pd} \rangle \end{bmatrix}$$

其中: N_p 表示所有物元对象的全体, $Y_{p1}, Y_{p2}, \dots, Y_{pd}$ 分别是集合 N_p 关于特征 c_1, c_2, \dots, c_d 的量值范围,即为节域。全体病人的节域为:

$$R_j = \begin{bmatrix} \text{全体病人} & \text{年龄} & \langle 18, 30 \rangle \\ & \text{身高} & \langle 160, 178 \rangle \\ & \text{体重} & \langle 45, 75 \rangle \\ & \text{入院天数} & \langle 3, 15 \rangle \end{bmatrix}$$

定义 4 关联函数。作为对物元对象进行聚类的规则,即把“具有性质 A ”的事物从定性描述扩展到“具有性质 A 的程度”的定量描述。设 $X_0 = \langle a, b \rangle, X = \langle c, d \rangle$, 且 $X_0 \subset X$, 关联函数为:

$$k(x) = \begin{cases} \frac{\rho(x, X_0)}{D(x, X_0, X)} - 1, & \rho(x, X_0) = \rho(x, X) \text{ 且 } x \notin X_0 \\ \frac{\rho(x, X_0)}{D(x, X_0, X)}, & \text{其他} \end{cases}$$

其中 $\rho(x, X_0) = \left| x - \frac{a+b}{2} \right| - \frac{b-a}{2}$ 为点 x 与区间 X_0 之距。区间 $\langle a, b \rangle$ 既可为开区间,也可可为闭区间,也可可为半开半闭区间。

$$D(x, X_0, X) = \begin{cases} \rho(x, X) - \rho(x, X_0), & \rho(x, X) \neq \rho(x, X_0) \text{ 且 } x \notin X_0 \\ \rho(x, X) - \rho(x, X_0) + a - b, & \rho(x, X) \neq \rho(x, X_0) \text{ 且 } x \in X_0 \\ a - b, & \rho(x, X) = \rho(x, X_0) \end{cases}$$

$$\text{其中 } \rho(x, X) = \left| x - \frac{c+d}{2} \right| - \frac{d-c}{2}.$$

$k(x) \geq 0$ 表示 x 属于区间 X_0 ; $k(x) < 0$ 表示 x 不属于区间 X_0 。关联函数表示物元对象关于簇中各特征属性的关联程度。区间 X_0 对应簇的经典域, X 对应全体物元对象的节域。

对于病人 6, 其物元表示为:

$$\begin{bmatrix} \text{病人 6} & \text{年龄} & 19 \\ & \text{身高} & 168 \\ & \text{体重} & 60 \\ & \text{入院天数} & 4 \end{bmatrix}$$

此病人关联函数的计算过程如下: 对于属性年龄 $x = 19$ 来说, 在簇 1 的经典域物元模型中年龄属性的经典域 $X_0 = \langle 18, 20 \rangle$, 在全体病人的节域物元模型中年龄属性的节域 $X = \langle 18, 30 \rangle$, $\rho(x, X_0) = \left| 19 - \frac{18+20}{2} \right| - \frac{20-18}{2} = -1$, $\rho(x, X) = \left| 19 - \frac{18+30}{2} \right| - \frac{30-18}{2} = -1$, $\rho(x, X) = \rho(x, X_0)$,

$D(x, X_0, X) = 18 - 20 = -2, k(x) = \frac{-1}{-2} = 0.50$, 其他属性的计算类似, 对于簇 2 采用与簇 1 同样的计算方法, 现将各计算结果在表 2 中列出。

表 2 病人 6 关于各簇属性的关联函数值

簇标号	年龄	身高	体重	入院天数
1	0.50	-0.33	-0.25	0.33
2	-0.75	0.17	0.00	-0.86

定义 5 综合关联函数。若簇中对象的各特征 c_1, c_2, \dots, c_d 的权重系数为 $\lambda_1, \lambda_2, \dots, \lambda_d$, 且满足 $\sum_{j=1}^d \lambda_j = 1$, 则物元对象 O 关于各簇的综合关联函数为 $K_i(O) = \sum_{j=1}^d \lambda_j k(c_{ij}(O)) = \sum_{j=1}^d \lambda_j k(v_{ij})$ 。它表示物元对象属于某簇的程度。 $\max_{1 \leq i \leq m} \{K_i(O)\}$ 表示物元对象 O 关于某簇综合关联度最大, 可据此判断物元对象 O 属于该簇。

假设病人 6 的各属性权重均为 0.25, 则病人 6 对于簇 1 的综合关联系数为 $0.25 \times 0.50 + 0.25 \times (-0.33) + 0.25 \times (-0.25) + 0.25 \times 0.33 \approx 0.06$, 对于簇 2 的综合关联系数计算过程类似, 计算结果为 $-0.36, \max\{0.06, -0.36\} = 0.06$, 说明病人 6 与簇 1 的关系较密切, 应该划分至簇 1 中。

定义 6 微簇。由聚类特征 $(R_i, CF2', CF1', n_i)$ 来表示, 它是一个 4 维元组。其中, 第一维 R_i 表示各微簇内所包含的所有物元对象在 d 维属性上的经典域集合, 它是一个物元模型的阵列, 形式为定义 2 所描述, $CF2', CF1', n_i$ 的定义和文献 [3] 一致。

2 EXCluStream 算法

借鉴经典算法 CluStream 的思想, EXCluStream 算法由在线和离线两个部分构成, 在线部分快速接受输入的数据流, 使用初始微簇算法进行初始聚类, 并且基于修改的 micro-cluster 结构定时存储数据流的摘要信息。随着新数据点的到达该中间结果实时进行更新, 离线部分 macro-cluster 通过对在线部分保存的中间结果的再处理得到用户感兴趣的在不同时间范围内数据流的聚类结果。EXCluStream 算法采用金字塔时间型的时间窗口分级保存数据的摘要信息。

2.1 初始微簇算法

输入 任意数据集, 权重系数 λ_j 。

输出 m 个簇的集合。

执行步骤如下。

1) 将所有数据点初始化为物元模型。

2) 随机选择两个物元对象, 建立对应簇的经典域模型 R_i 与整个物元对象集的节域模型 R_p 。

3) 从剩余物元对象中任意选择一个对象, 对于每个簇用关联函数 $k(x)$ 计算该对象关于各个特征属性的关联度, 然后用权重系数 λ_j 计算出该对象对于各个簇的综合关联度 $K_i(O)$, 比较得出 $\max[K_i(O)]$ 。

4) 若 $\max[K_i(O)] < 0$, 则为其创建一个新簇; 否则取 $\max[K_i(O)]$ 所对应的簇来分配物元对象。

5) 调整各簇的经典域模型 R_i 与整个物元对象集上的节域模型 R_p 。如果所有物元对象处理完毕, 算法终止; 否则转步骤 3)。

这部分的细节如下:

Init-EXCluStream()

{ do

将所有数据点初始化为物元模型

While(未到数据末尾);

//将每个数据点的属性与其

//对应的值用物元的形式表示

随机选择两个物元对象, 建立初始经典域模型 R_i , 初始节域模型 R_p ;

//这时只存在一个簇, 且该簇的经典域等于节域

for(剩余的每一个物元对象)

{ for(每个簇)

{ 计算物元对象关于各属性的关联度;

根据权重系数 λ_j 计算综合关联度 $K_i(O)$;

//关联度的计算使用定义 4 中的关联函数确定,

//权重系数根据各属性的重要程度赋值

}

在所有的综合关联度中比较得出 $\max[K_i(O)]$;

if ($\max[K_i(O)] < 0$) 为该物元对象创建新簇;

// $K_i(O)$ 的取值大小代表了该物元属于此簇的程度;

else 分配该物元对象至 $\max[K_i(O)]$ 所对应的簇;

调整各簇的经典域模型 R_i 和节域模型 R_p ;

// 若物元对象由 $\max[K_i(O)]$ 判断出属于某簇, 则根据此

// 物元对象各属性的取值调整所分配的簇的经典域;

// 若由 $\max[K_i(O)] < 0$ 判断出物元对象不属于任何簇,

// 则为此物元对象建立新簇, 新簇的经典域为 $\max[K_i(O)]$

// 对应的簇的经典域边缘值与该物元对象的属性值之间

// 的范围。当某簇的经典域经调整后超过节域的范围或

// 有新簇建立时, 调整节域的值

2.2 在线层算法

输入 动态数据流, 权重系数 λ_j 。

输出 存储于金字塔模型中的微簇。

执行步骤如下。

1) 微簇的初始化。

接收一批流数据后将其存入硬盘, 用离线的方式进行微簇的初始化。此过程调用初始微簇算法。

2) 添加数据对象。

对于每一个新到达的数据对象, 将其初始化为物元模型并计算对于各个簇的综合关联度 $K_i(O)$ 。当 $\max[K_i(O)] < 0$ 时, 为其建立新簇, 初始化新簇的经典域模型和簇聚类特征, 修改整个物元对象的节域模型 R_p ; 否则根据 $\max[K_i(O)]$ 将其归入某簇, 修改微簇聚类特征 $(R_i, CF2', CF1', n_i)$ 。

3) 删除与合并微簇策略。

当产生新微簇时, 有可能要删去旧微簇或者合并两个已有微簇以节省内存空间。采用的策略是删去最近最少使用的微簇, 否则合并两个经典域范围跨度最小且彼此范围邻接的簇合并。

4) 存储快照。

在步骤 2) ~ 3) 进行的同时, 根据金字塔时间框架的要求实时将微簇的快照存入硬盘中。

在线层算法的细节如下:

Online-EXCluStream()

{ 接收一批数据流, 将其存入硬盘;

调用初始微簇算法;

for(每一个新到达的数据对象)

{ 初始化为物元模型;

for(每个簇)

{ 计算关联度;

根据权重系数 λ_j 计算综合关联度, 比较得出 $\max[K_i(O)]$;

}

if ($\max[K_i(O)] < 0$)

{ if (内存不足)

{ if (存在某微簇最近最久未使用)

删去该簇;

```

else 合并范围最小且彼此邻接的两个经典域  $R_i$  和  $R_{i+1}$ ;
}
else { 为新对象建立新簇;
      初始化新簇的经典域模型和簇聚类特征, 修改节域模型;
}
}
else { 根据  $\max[K_i(O)]$  将其归入对应簇;
      修改簇聚类特征( $R_i, CF2^i, CF1^i, n_i$ );
}
}
}
将微簇按照金字塔时间要求存入硬盘;
}

```

2.3 离线层算法

使用修改的初始微簇算法对用户指定时间范围内的微簇进行再聚类, 返回聚类结果。

输入 时间范围 horizon。

输出 用户指定时间范围内的聚类结果。

执行步骤如下。

1) 从磁盘上取出时间范围 horizon 内的所对应的微簇, 选取包含足够多数目物元对象的微簇作为初始种子簇。

2) 对于剩余的每一个微簇, 将此微簇看作为一个虚拟点, 计算其经典域的中点。

3) 根据中点值计算虚拟点对各簇的综合关联度, 取最大综合关联度对应的簇分配此虚拟点或为其建立新簇。

4) 调整各簇的经典域和整个物元对象集的节域。如果所有虚拟点都已处理完毕, 则算法结束; 否则转步骤 2)。

离线层算法与初始微簇算法类似, 在此不再赘述伪代码, 仅将这两个算法的区别列出如下:

1) 初始的时候, 种子不再是随机选择的两个物元对象, 而是选取两个包含足够多数目物元对象的微簇;

2) 划分的时候, 对于未知归属的微簇来说, 将此微簇看作为一个虚拟点, 即取经典域的中点值来计算虚拟点关联度, 而不直接取物元对象的各特征值来计算关联度。

3 实验验证

实验环境 Intel 奔腾双核 T2390 处理器, 主频 1.86 GHz, 1 GB 内存, 操作系统为 Windows XP, 算法用 C++ 实现。

实验数据采用真实数据集和人工数据集。真实数据集采用 KDD-CUP99 网络入侵检测数据集, 它是麻省理工学院林肯实验室连续两个星期的网络流检测记录。该数据集共包含 494 020 条 TCP 连接记录, 每条记录包含 41 维属性, 连续属性 34 维, 标称属性 7 维。本文采用和文献[3]相同的方法, 只使用其中 34 条连续属性。CluStream 算法是数据流挖掘的经典算法, 而文献[6]中的 ACluStream 算法是针对前者提出的较有影响力的改进算法, 因此, 本文采用这两个算法作为对比算法, 来验证 EXCluStream 算法的有效性。

人工数据集是通过 Matlab 软件生成的, 且满足一系列高斯分布。采用以下含义来命名该数据集: B 表示数据集中元组个数, C 表示自然簇个数, D 表示每个元组的维度。

3.1 算法的聚类质量

由于平方距离和 (Sum of Square Distance, SSQ) 只适用于评价球状簇, 对于发现任意形状簇的评价效果不佳, 所以本文选择文献[4]中使用的聚类纯度作为衡量算法性能的标准。聚类纯度定义为每个聚类结果中真实主导类别所占比例的平均值。

实验数据采用 KDD-CUP99 数据集, 三种算法在代表性时间戳上的实验结果如图 1 所示, 这些代表性时间戳均为文献

[4] 中采用的数据。可以看到, 在代表时间戳 433, 1 857 上, EXCluStream 算法的聚类纯度要远远高于其他两种算法, 表现出更好的聚类质量, 而 CluStream 算法的聚类纯度不到 80%, ACluStream 算法的聚类纯度不到 90%。这是由于在 EXCluStream 算法中采用了关联函数值来代替传统聚类算法中用于度量相似度的距离, 不使用平方误差准则和绝对密度作参数, 因此数据点能够更自然地分配到所属簇当中, 所以该算法相对于其他两种算法有更准确的划分结果。

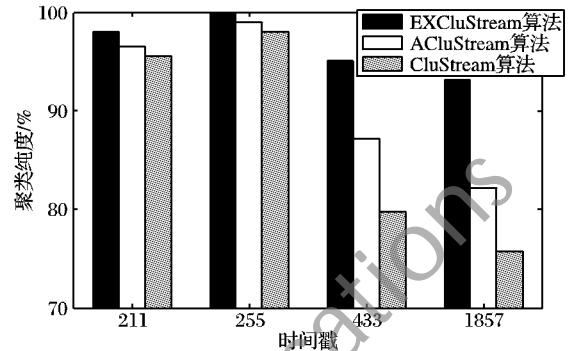


图1 聚类纯度比较 (horizon = 1 s, 流速为每秒 200 个数据点)

3.2 算法的有效性

有效性是衡量数据流算法的重要指标, 本文使用数据流处理速率来测量 EXCluStream 算法的有效性, 实验数据采用 KDD-CUP99 数据集, 实验结果如图 2 所示。可以看出, 在算法开始时, 三者的处理速率比较接近, EXCluStream 算法比 CluStream 算法略低些, 比 ACluStream 算法略高些, 但随着时间的流逝, EXCluStream 表现出更好的性能, 能够在单位时间段里处理更多的数据点。这是因为该算法在迭代执行的过程中不需要计算准则函数, 更新微簇时对于聚类特征的计算也较简单, 因此大大节省了处理时间。

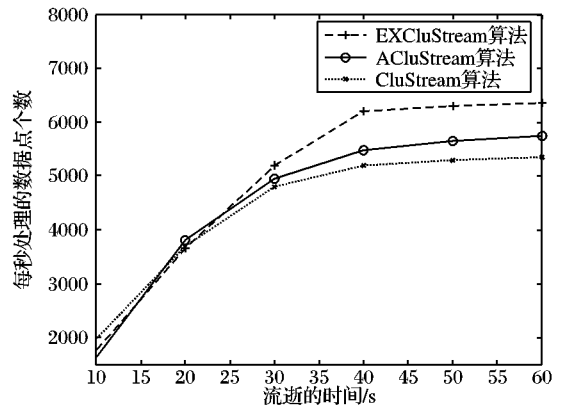


图2 处理速率比较 (流速为每秒 2000 个数据点)

3.3 算法的可扩展性

采用人工数据集 B400C20、B200C10 和 B100C5 来测试算法在维度上的可扩展性, 如图 3 所示; 采用人工数据集 B400D40、B200D20 和 B100D10 来测试算法在簇数目上的可扩展性, 如图 4 所示。从图中可以看出, 随着维度和簇数目的增长, 各个数据集的处理时间基本都呈线性增长, 说明 EXCluStream 算法比较稳定, 能够适应不同规模的数据集。通过分析本文算法的时间复杂度可以说明原因: EXCluStream 算法的基本操作是计算每一个数据点对于当前各簇的综合关联度, 而综合关联度的计算又依赖于数据点的维度, 因此初始微簇算法的时间复杂度为 $O(nld)$, 其中: n 为数据点个数, l 为当前簇数目, d 为数据点维数。在线与离线的过程都运用了初始微簇算法, 基本操作类似, 故 EXCluStream 算法的在线与离线过程的时间复杂度均为 $O(nld)$ 。

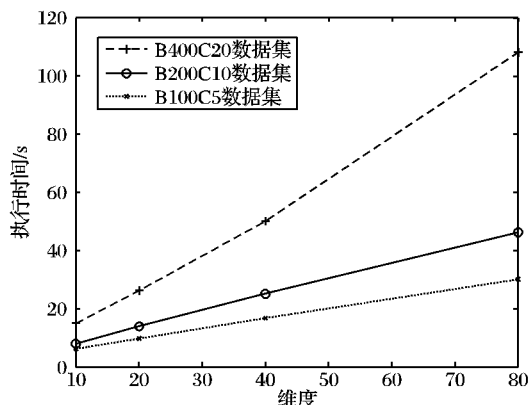


图3 在维度上的可扩展性(流速为每秒100个数据点)

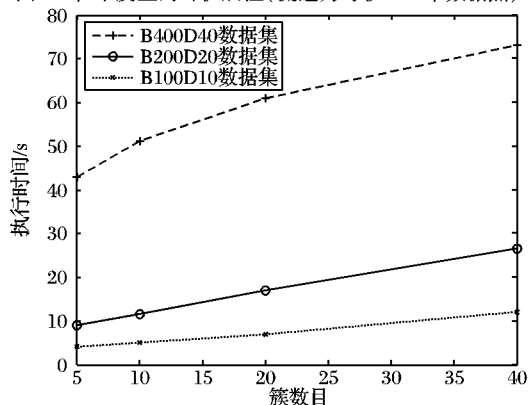


图4 在簇数目上的可扩展性(流速为每秒100个数据点)

4 结语

虽然 CluStream 算法所提出的框架对数据流的聚类具有很大的意义,但该算法聚类效果不佳,在本文所提的算法中,用数据点进行物元化后的关联函数值来代替传统的衡量数据点归属的距离指标,很好地解决了 CluStream 算法中存在的问题。但是,由于要存储各簇的经典域和节域,当自然簇数目较多时,该算法的空间复杂度会随着自然簇的增多而呈线性增长,因此,如何对自然簇数目较多的数据流进行有效聚类,将是下一步要解决的问题。

参考文献:

[1] GUHA S, MISHRA N, MOTWANI R, *et al.* Clustering data

streams [C]// Proceedings of the 41st Annual Symposium on Foundations of Computer Science. Washington, DC: IEEE Computer Society, 2000: 359–366.

- [2] O'CALLAGHAN L, MISHRA N, MEYERSON A, *et al.* Streaming-data algorithms for high-quality clustering [C]// Proceedings of the 18th International Conference on Data Engineering. Washington, DC: IEEE Computer Society, 2002: 685–694.
- [3] AGGARWAL C C, HAN J, WANG J, *et al.* A framework for clustering evolving data streams [C]// Proceedings of the 29th International Conference on Very Large Data Bases. Berlin: VLDB Endowment, 2003: 81–92.
- [4] AGGARWAL C C, HAN J, WANG J, *et al.* A framework for projected clustering of high dimensional data streams [C]// Proceedings of the 30th International Conference on Very Large Data Bases. Toronto: VLDB Endowment, 2004: 852–863.
- [5] CAO F, ESTER M, QIAN W N, *et al.* Density-based clustering over an evolving data stream with noise [C]// Proceedings of the 6th SIAM International Conference on Data Mining. Philadelphia: SIAM Press, 2006: 328–339.
- [6] 朱蔚恒, 印鉴, 谢益煌. 基于数据流的任意形状聚类算法[J]. 软件学报, 2006, 17(3): 379–387.
- [7] 张晨, 金澈清, 周傲英. 一种不确定数据流聚类算法[J]. 软件学报, 2010, 21(9): 2173–2182.
- [8] 王述云, 胡运发, 范毅捷, 等. 基于距离与熵的混合属性数据流聚类算法[J]. 小型微型计算机系统, 2010, 31(12): 2365–2371.
- [9] 蔡文, 杨春燕, 陈文伟, 等. 可拓集与可拓数据挖掘[M]. 北京: 科学出版社, 2008: 39–43.
- [10] LI Q X. The interval elementary dependent function based on interval side-distance [C]// Proceedings of IEEE 2008 ISECS International Colloquium on Computing, Communication, Control, and Management. Washington, DC: IEEE Computer Society, 2008: 674–678.
- [11] 李桥兴, 刘思峰. 基于区间距和区间侧距的初等关联函数构造[J]. 哈尔滨工业大学学报, 2006, 38(7): 1097–1100.
- [12] 李桥兴. 一元多维位值公式及一元多维初等关联函数构造方法[J]. 兰州大学学报: 自然科学版, 2010, 46(2): 86–90.
- [13] 杨春燕, 蔡文. 基于可拓集的可拓分类知识获取研究[J]. 数学的实践与认识, 2008, 38(16): 184–191.
- [14] 蔡文, 杨春燕. 可拓学的应用研究、普及与推广(综述)[J]. 数学的实践与认识, 2010, 40(7): 214–220.

(上接第201页)

缓存来生成一个更普遍的引力函数来减少移动代价。仿真实验表明该算法的有效性。但该算法的启发引力函数还没达到最优,以后的工作还需对其改进,以使启发引力函数能更高效地指引树的生长。该算法只是在理论上得到了实现,还未能应用在实际的双足机器人上,这也将是以后努力的方向。

参考文献:

[1] 夏泽洋, 陈恩, 熊璟, 等. 仿人机器人运动规划研究进展[J]. 高技术通讯, 2007, 17(10): 1092–1099.

[2] FU C L, CHEN K. Gait synthesis and sensory control of stair climbing for a humanoid robot [J]. IEEE Transactions on Industrial Electronics, 2008, 55(5): 2111–2120.

[3] 石为人, 黄兴华, 周伟. 基于改进人工势场法的移动机器人路径规划[J]. 计算机应用, 2010, 30(8): 2021–2023.

[4] JEAN F. Complexity of nonholonomic motion planning [J]. International Journal of Control, 2001, 74(8): 776–782.

[5] LaVALLE S M. Motion planning: the essentials [J]. IEEE Robotics and Automation Society Magazine, 2011, 18(1): 79–89.

[6] 康亮, 赵春霞, 郭剑辉. 未知环境下改进的基于 RRT 算法的移动机

人路径规划[J]. 模式识别与人工智能, 2009, 22(3): 337–343.

- [7] LAVALLE S M, KUFFNER Jr, J J. Rapidly-exploring random trees: progress and prospects [C]// Proceedings of the Fourth Workshop on the Algorithmic Foundations of Robotics. Natick, MA: A K Peters, 2000: 45–59.
- [8] 张振荣, 刘惊雷, 张伟. 一种生成最优联盟结构的任意时间算法[J]. 计算机工程, 2011, 37(2): 185–187.
- [9] 郭亚军, 鲁汉榕. 任意时间算法的性能描述[J]. 武汉交通科技大学学报: 交通科学与工程版, 2000, 24(5): 562–565.
- [10] URMSOON C, SIMMONS R. Approaches for heuristically biasing RRT growth [C]// IROS 2003: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE Press, 2003: 1178–1183.
- [11] 夏泽洋, 陈恩. 仿人机器人足迹规划建模及算法实现[J]. 机器人, 2008, 30(3): 231–237.
- [12] 胡金东, 刘国栋. 双足机器人步行模式的在线全身修正[J]. 计算机应用, 2011, 31(1): 286–288, 292.
- [13] 梶田秀司. 仿人机器人[M]. 管贻生, 译. 北京: 清华大学出版社, 2007.