第35卷 第14期 计 算 机 工 程 2009年7月
Vol.35 No.14 Computer Engineering July 2009
·软件技术与数据库· 文章编号：1000—3428(2009)14—0049—03 文献标识码：A 中图分类号：TP311

多数据流的增量聚类实现与应用

张锡琴
(浙江工业大学经贸管理学院，杭州 310032)

摘 要：针对时间序列数据流的增量聚类研究较少的现状，采用多维时态子空间聚类对数据流的增量聚类进行探究。多维时态子空间聚类是指在连续一段时间内，数据流中的值的距离小于2α，它的另一个要求是最后的聚类结果必须包含一定数量的数据流。聚类结果随时间的演变能持续增量地更新，这个更新机制采用滑动窗口的形式，把最早时刻的数据删除后，添加入新到达的数据。采用股票数据对算法进行测试与验证，实验证明，该算法效果较好。

关键词：数据流；增量聚类；多维时态子空间聚类

Realization and Application of Incrementally Clustering of Multi-data Streams

ZHANG Xi-qin
(College of Business Administration, Zhejiang University of Technology, Hangzhou 310032)

【Abstract】 There is only a little research on incrementally cluster temporal data streams. This paper focuses on the problem of clustering temporal data streams based on the sliding window. Temporal Multiple-dimension Subspace α-Cluster(TMSC) is adopted. It consists of data streams, whose distance is less than 2α, and must contain a number of streams which are predefined. The result of the clustering is updated through time evolving. Sliding window mechanism is adopted to realize it. The earliest data is removed, and the new coming data is added. Stock data is used to test the algorithm, and the result is quite well.

【Key words】 data stream; incrementally clustering; Temporal Multiple-dimensional Subspace α-Cluster(TMSC)

1 概述

所谓数据流就是大量连续到达的、潜在无限的数据的有序序列，这些数据只能按照顺序存取并被读取一次或有限次[1]。如股票数据、某个地方的温度数据等。

由于数据流是不断到来的，每时每刻的数据都可能不同，因此必须采用增量聚类算法来处理。所以数据挖掘应该是一个在线的、连续的过程，而不是随机的过程[2]。

目前关于数据流挖掘的大多数聚类算法都是针对单个数据流内部进行聚类，对多数据流之间的聚类文章少。本文针对的就是多数据流之间的聚类。因为大多数算法对于数据流中的时间未有严格的定义，所以本文引入了时态的概念。

已存在很多数据流的挖掘，都是把整个滑动窗口内部的一条数据流作为聚类对象进行聚类。也就是说，一旦滑动窗口确定，小于滑动窗口这个时间段之内的相似类可能就无法发现。这样的聚类算法意义不大。因此，应该查找一个更好的聚类方法。本文在时态的基础上[3]，修改和完善了文献[4]的算法，提出了多维时态子空间聚类算法(Temporal Multiple-dimension Subspace α-Cluster, TMSC)。

2 问题描述和相关概念

为了说明把数据流中最后 W 个值用一个矩阵来表示，表1列出了本文中的主要符号以及它们的定义。

下面是本文中使用的基本定义。

定义 1 设 μ 是从 ATT t 到 ATI $\mu(t)$ 的映射，也即 $R\to 2^R$，如果 μ 满足下列性质，则称 μ 为时态型，$\mu(t)$ 为 μ 的时态因子[3]：

(1)非空性。$t\in\mu(t)$。

(2)单调性。若 $t_1<t_2$ 且 $\mu(t_1)\cap\mu(t_2)=\varnothing$，$\forall t'\in\mu(t_1)$ 和 $\forall t''\in\mu(t_2)$，$t'<t''$，记作 $\mu(t_1)<\mu(t_2)$。

(3)同一性。$\forall t'\in\mu(t),\ \mu(t')=\mu(t)$。

(4)有界性。$\forall t'\in\mu(t),\ |t'|<+\infty$。

表1 本文中的基本标志

符号	描述
s,s_i	时态数据流
$s[i]$	第$[i]$个时刻的数据的值
N	数据流的数量
W	滑动窗口的大小
C_i	最大多维时态子空间的 α-*cluster*
$c_{i,j}$	i 时刻的第 j 个简单 α-*cluster*
c	简单 α-*cluster*
m	一个 *cluster* 中的数据流的数量
G,G_i	候选的简单 α-*cluster* 组
minRows	聚类中所必须包含的数据流的数量
minCols	聚类中至少连续的时间跨度
2α	2个数据流之间最大的距离

显然，时态型 μ 是对时间数轴 R 的一个划分，每个时态因子 $\mu(t)$ 是一个 ATT 集合。秒、分、小时、日、周、月和年等可以用来划分时间数轴 R，并且它们都满足上述性质，因此都是时态型。

定义 2(时态数据流) 设 S 表示一条时态数据流，v 是一

作者简介：张锡琴(1984－)，女，硕士研究生，主研方向：时态数据挖掘

收稿日期：2009-03-04 **E-mail：**yayoumeigirl@163.com

个时态型，记 $S=\{(v(t_1),A_1),(v(t_2),A_2),\cdots,(v(t_m),A_m),\cdots\}$ 且 $t_1<t_2<\cdots<t_m$。每个对象 A_m 含有 p 个属性 $e_1,e_2,\cdots,e_p$，每个属性 $e_i(i=1,2,\cdots,p)$都是数值型或者是可以转换为数值型的。

定义 3(简单 *α-cluster*) 简单 *α-cluster* 包含了一些数据流，这些数据流在相同时刻的值的聚类是小于等于 2α 的。每个类中包含的数据流的数量是没有严格限制的。

定义 3 中显示的就是本文中对聚类的差异度标准，也就是在算法中，在同一个时刻，认为距离小于等于 2α 的就属于同一个类别。

为了得到简单的 *α-cluster*，采取如下所示的方法。

输入 data stream *S*, *α*

输出 simple *α-cluster*: *list*

```
For i=1 to N
 For j=i+1 to N
  计算距离 D=DISTANCE(Si, Sj)
  初始化当前聚类
  if D<α then
   add Sj to the current cluster
  End if
 End for
 If not exists in the list
  Add to the list
 End if
End for
```

定义 4 (时态子空间 *α-cluster*) 时态子空间 *α-cluster* 是指必须包含 *minRows* 数据流，在这些数据流中，在每个时间点上最大的差别应小于等于 2α，并且必须跨越连续的 *minCols* 个时间段。

时态子空间 *α-cluster* 是由$(S,[d_i,d_j])$表示的，S 是由数据流组成的，$[d_i,d_j]$是指连续的时间段，$i\le j$。

定义 5(最大时态子空间 *α-cluster*) 如果说$(S,[d_i,d_j])$是最大的，那么将找不到$(S,[d_k,d_l])$, $k\le i$ 并且 $l\ge j$，也找不到$(T,[d_i,d_j])$, $S\subset T$。

定理 1(闭包属性) 如果 $C=(S,[d_i,d_j])$是一个时态子空间 *α-cluster*，不一定是要最大时态子空间 *α-cluster*，然后每一个 $C'=(S',[d_k,d_l])$，$S'\subset S$，$k\ge i,l\le j$，$|S'|\ge minRows$，$l-k+1\ge minCols$，那么 C' 也是一个时态子空间 *α-cluster*。

证明：假设 C' 不是子空间 *α-cluster*，那么 C 不可能成为子空间 *α-cluster*。

由闭包属性，没有必要查找所有的子空间 *α-cluster*，只要找到它们的子集即可。

定理 2(距离属性) 如果 $C_1=([S_1,S_2,\cdots,S_K,S_{K+1}],[d_m,d_n])$, $C_2=([S_1,S_2,\cdots,S_K,S_{K+2}],[d_k,d_l])$和 $C_3=([S_{K+1},S_{K+2}],[d_i,d_j])$是时态子空间 *α-cluster*，那么 $C=([S_1,S_2,\cdots,S_K,S_{K+1},S_{K+2}],[d_e,d_f])$也是时态子空间 *α-cluster*，$d_e=\max\{d_m,d_k,d_i\}$, $d_f=\min\{d_n,d_l,d_j\}$。

证明：从定理 1 的闭包属性中可以得到 $C_1'=([S_1,S_2,\cdots,S_K,S_{K+1}],[d_e,d_f])$, $C_2'=([S_1,S_2,\cdots,S_K,S_{K+2}],[d_e,d_f])$以及 $C_3'=([S_{K+1},S_{K+2}],[d_e,d_f])$都是时态子空间 *α-cluster*。所以 S_{K+2}和 $S_1,S_2,\cdots,S_K,S_{K+1}$ 之间的距离都是小于 2α，那么 $C=([S_1,S_2,\cdots,S_K,S_{K+1},S_{K+2}],[d_e,d_f])$也是时态子空间 *α-cluster*。

3 算法框架及描述

3.1 聚类初始化阶段

聚类初始化阶段是根据时间序列的最后 W 个值确定最初的最大时态子空间 *α-cluster*。一个聚类必须包含最少 *minRows* 数据流和每个数据流必须包含 *minCols* 个连续的时间段。

初始化阶段可以分为多个阶段：第 1 步，查看每一个时间点的值，以此来确定简单 *α-cluster*。第 2 步，算法产生了 m=2 个数据流的最大 *Cols*(最长时间相似)，接下来就是不断尝试 $m=m+1$，直到找到最大的时态子空间 *α-cluster*。如果聚类结果中包含的时间跨度小于 *minCols*，那么丢弃这个聚类结果。

下面说明聚类初始化阶段。表 2 是原始的数据流。

表 2 数据流中的值

	d_1	d_2	d_3	d_4
s_1	7.4	4.6	6.7	5.5
s_2	6.4	3.7	8.2	3.7
s_3	8.1	3.9	8.6	5.6
s_4	5.2	6.0	5.5	5.8
s_5	8.0	3.2	7.8	8.3

表 3 展示了每一个时间刻度上面的聚类。表 4 展示了 2-level 的聚类。2-level 聚类是根据简单 *α-cluster* 得到的。简单 *α-cluster* 放在表 4 的第 5 列。候选的 2-level 聚类被分成 4 个组，分别为 $G1,G2,G3,G4$。每一组中的候选的聚类必须包含 m-1 个以上的聚类。每一个组都是分别处理的，从第 1 个组开始处理。

表 3 简单 *α-cluster*

	d_1	d_2	d_3	d_4
s_1	$c_{1,2}$	$c_{2,1},c_{2,2}$	$c_{3,1},c_{3,2}$	$c_{4,1}$
s_2	$c_{1,1},c_{1,2}$	$c_{2,1}$	$c_{3,2}$	$c_{4,2}$
s_3	$c_{1,2}$	$c_{2,1}$	$c_{3,2}$	$c_{4,1}$
s_4	$c_{1,1}$	$c_{2,2}$	$c_{3,1}$	$c_{4,1}$
s_5	$c_{1,2}$	$c_{2,1}$	$c_{3,2}$	$c_{4,2}$

表 4 候选的 2-level *α-cluster*

组	No	流	时间	聚类结果
*G*1	1	s_1,s_2	d_1~d_3	$c_{1,2}\|c_{2,1}\|c_{3,2}$
	2	s_1,s_3	d_1~d_4	$c_{1,2}\|c_{2,1}\|c_{3,2}\|c_{4,1}$
	3	s_1,s_4	d_2~d_4	$c_{2,2}\|c_{3,1}\|c_{4,1}$
	4	s_1,s_5	d_1~d_3	$c_{1,2}\|c_{2,1}\|c_{3,2}$
*G*2	5	s_2,s_3	d_1~d_3	$c_{1,2}\|c_{2,1}\|c_{3,2}$
	6	s_2,s_4	d_1	$c_{1,1}$
	7	s_2,s_5	d_1~d_4	$c_{1,2}\|c_{2,1}\|c_{3,2}\|c_{4,2}$
*G*3	8	s_3,s_4	d_4	$c_{4,1}$
	9	s_3,s_5	d_1~d_3	$c_{1,2}\|c_{2,1}\|c_{3,2}$
*G*4	10	s_4,s_5	-	-

定理 3(聚类剪枝标准)[4] 如果某一个组的候选的 m-level 聚类少于 *minRows*-m+1，那么该组可以安全删除了。

定理 4(时间剪枝)[4] 如果在一个组中的候选的 *α-cluster* 包含 m 个数据流，并且某一个时间刻度上出现的次数小于 *minRows*-m+1，那么这个时间刻度也可以删除了。

尝试从通过剪枝的类中生成包含 m+1 个数据流的聚类。融合第 1 个和第 2 个数据流、第 1 个和第 4 个数据流、第 2 个和第 4 个数据流(第 3 个数据聚类已经被删除)，得到表 5。

表 5 level=3 聚类结果

No	数据流	时间段	简单 *α-cluster*
1	s_1,s_2,s_3	d_1~d_3	$c_{1,2}\|c_{2,1}\|c_{3,2}$
2	s_1,s_2,s_5	d_1~d_3	$c_{1,2}\|c_{2,1}\|c_{3,2}$
3	s_1,s_3,s_5	d_1~d_3	$c_{1,2}\|c_{2,1}\|c_{3,2}$

仔细检查这 3 个数据流聚类结果，很明显 No1 和 No2 是不符合剪枝标准的，第 1 个和第 2 个聚类可以组合成表 6 中的样子。

表 6 最后的聚类结果

No	流	时间	简单 *α-cluster*
1	s_1,s_2,s_3,s_5	d_1~d_3	$c_{1,2}\|c_{2,1}\|c_{3,2}$

定理 5(流剪枝策略)[4] 如果剩下的数据流的个数少于 *minRows*，那么所有由这些数据流组成的组都可以被删除，因为它们不可能产生子空间 *α-cluster*。

这个算法的主要框架可以表示如下：

```
初始化聚类(S, α, minRows, minCols, W)
Input S: set of streams,
      α：同一个时间上的同一个聚类中的最大差别,
      minRows: 每一个聚类中最小的数据流,
      minCols: 最短的连续时间,
      W:滑动窗口的大小
Output A：最大子空间 α-cluster
for   i=1 to W
  计算每一个时间点的所有简单的 α-cluster
end for
for i=1 to N-minRows+1
  set m=2;
  产生 m-level 候选的 α-cluster;
  应用聚类剪枝和维度剪枝
  while there exist m-level 候选结果 do
   产生 m+1-level 候选的 α-cluster;
   m=m+1;
   if m>=minRows and C is 最大时态子空间 α-cluster then
    更新 A;
   end if
   应用聚类剪枝和时间剪枝
  end while
end for
report A
```

3.2 聚类保持

聚类保持(Cluster Maintenance, CM)阶段就是让聚类结果始终都是最新的，它必须考虑到新的数据的到达。这个阶段就是当新的值到达值，增量聚类。

在这种情况下，当新的时刻来临时，所有的数据流都更新了。因为算法处理是基于滑动窗口机制，所以滑动窗口最左边的值将被删除，新的值将增加进来。

图 1 展示了新的值到来的状况，因此第 1 个维度的简单 *α-cluster* 需要被删除。

	~~d_1~~	d_2	d_3	d_4	d_5
s_1	~~7.4~~	4.6	6.7	5.5	8.2
s_2	~~6.4~~	3.7	8.2	8.0	6.1
s_3	~~8.1~~	3.9	8.6	5.6	8.3
s_4	~~5.2~~	6.0	5.5	5.8	5.5
s_5	~~8.0~~	3.2	7.8	8.3	5.0

图 1 新时刻 d_5 的到来

刚开始，检查最大子空间 *α-cluster*，因为 d_1 这个时间的删除，可能某一些最大子空间 *α-cluster* 需要删除。另外，因为 d_5 的到来，有些存在的聚类可能会扩展。

然后，算法查找新的最大子空间 *α-cluster*。

需要检查每一个包含 d_4，并且作为其最右边的时间的聚类，因为它们可能可以加上 d_5 这个时间。如果这个类可以被扩展，那么它就可以包含在答案当中。接下去，从所有包含 d_1 的聚类中删除 d_1。如果删除了 d_1 后，剩下的时间少于 *minCols*，那么这个类将被删除。最后，其他的聚类都不会被 d_1 的删除和 d_5 的加入受到影响。

4 数值实验结果及分析

这里采用股票数据进行聚类。股票数据是采用了一个小程序从同花顺股票软件中下载的历史数据中取得的数据。上海的数据为 852 个数据流，深圳的数据为 669 个数据流，起始时间为 2007 年 1 月 1 日到 2007 年 12 月 31 日。采用的滑动窗口大小为 100。以每只股票的收盘价为标准进行聚类。

表 7 显示的是不同的 α，*minRows*(用 R 来代替)和 *minCols*(用 C 来代替)下的初始化时间，30 次更新操作和 70 次更新操作以后的聚类数量和平均更新时间。因为作为属于同一个类别的判断标准是两者的距离小于 α，所以无需验证聚类的准确性，只需要查看该算法的时间复杂度即可。

表 7 股票数据的平均更新时间和聚类结果

α	R	C	初始化		30 次更新操作后		70 次更新操作后	
			聚类	时间/s	聚类	时间/s	聚类	时间/s
0.1	11	2	240	5	220	0.1	225	0.1
0.2	15	4	160	14	153	0.3	155	0.8
0.3	20	2	300	60	289	0.5	293	1.9
0.4	30	2	140	300	138	1.5	146	26.1

因为算法是针对时态数据的算法，所以必须符合时态数据的特性。为了保证算法的有效性，必须对算法进行时态转换，这一部分转换就是把所有的股票开盘日映射到连续的时态因子上。

本程序采用 Eclipse 进行开发，数据库采用 SQL Server, 内存为 1 GB。

5 结束语

本文针对时态数据流的特点，采用滑动窗口和 *α-cluster*，提出了针对时态数据的子空间 *α-cluster* 算法。该算法简单易实现，针对 2007 年 1 月 1 日~2007 年 12 月 30 日一年的数据进行聚类测试，以其收盘价作为数据进行测试，聚类算法能发现价格类似的股票，说明算法是有效的。

参考文献

[1] 蒋盛益, 李庆华, 李 新. 数据流挖掘算法综述[J]. 计算机工程与设计, 2005, 26(5): 1130-1132, 1169.

[2] Yang Qiang, Wu Xindong. Challenging Problems in Data Mining Research[J]. International Journal of Information Technology & Decision Making, 2006, 5(4): 597-604.

[3] 孟志青. 时态数据采掘中的时态型与时间粒度研究[J]. 湘潭大学自然科学学报, 2009, 22(3): 1-4.

[4] Maria K, Apostolos N P, Yannis M. Continuous Subspace Clustering in Streaming Time Series[J]. Information Systems, 2008, 33(2): 240-260.

编辑 顾逸斐