在数据流数据库中集成聚类算法研究与实现

张 晶，张 阳†
（西北农林科技大学 信息工程学院，陕西 杨凌 712100）

摘 要：在数据流管理系统中集成数据挖掘功能，有助于对数据流进行更加有效的管理和挖掘，但目前研究界对此方面工作关注不够。基于数据流管理系统 Esper，利用时间窗口和自定义函数，采用 Esper 处理语言改写 Clustream 算法，在 Esper 系统中实现聚类算法。实验结果表明，该方法可以 Esper 具有对数据流进行聚类分析的能力；与用 Java 实现数据流聚类相比，在 Esper 中实现聚类方法具有更好的处理多维大数据量数据流的能力。

关键词：数据流数据库；Esper 系统；时间窗口；Clustream 算法
中图分类号：TP315 **文献标志码**：A **文章编号**：1001-3695(2014)05-1456-03
doi:10.3969/j.issn.1001-3695.2014.05.041

Programming of micro-clusters algorithm in data stream management system

ZHANG Jing, ZHANG Yang†
(*College of Information Engineering, Northwest A&F University, Yanglin Shaanxi* 712100, *China*)

Abstract: To integrate data mining functions inside data stream management system is helpful for manage and mine data stream more efficiently. However, currently, the research community has not paid enough attention to this research field. Based on Esper, a data stream management system, this paper utilized time window and user defined function, and re-wrote the Clustream algorithm by Esper processing language, so as to realize clustering algorithm inside Esper. The experiment result shows that this way can implement the ability of clustering data streams in Esper. Compared with to implement data stream clustering in Java, to implement it in Esper can assure better clustering ability.

Key words: DSMS; Esper; time window; Clustream algorithm

0 引言

数据流是连续、有序、快速变化、没有终点、海量的数据[1]，是现在的研究热点之一。DSMS(data stream management system，数据流管理系统)是能够对动态的、时效性、实时性、无限性、瞬间性的数据进行预定性、连续性、概要性的查询处理的，能够满足海量信息的有效存储及处理的新型的数据管理系统[2,3]，但直接用 DSMS 处理复杂问题的效果仍然不太理想[4]。

因此，在 DSMS 中实现数据挖掘功能，对数据库及数据挖掘技术的研究者而言是一项新的挑战。在 DSMS 中嵌入数据挖掘算法则是数据库技术研究与发展的必然趋势[5]，而集成算法的 DSMS 则有一般数据库没有的优势：a)集成算法的 DSMS 比传统的扩展性更好，最重要的是使之具有了数据挖掘功能，以便用户更好地对数据流进行分析和处理；b)无论是否有数据挖掘专业知识，用户都可直接通过集成算法的 DSMS 处理数据，而且在存储管理、容错能力和访问安全上都拥有较大优势[6]。

在 DSMS 中嵌入数据挖掘算法，主要是利用其原有的机制，将需要用到的算法转换成为能够被系统识别的查询语句，使之在数据库中轻松运用算法来处理数据。目前，该方面只有在数据流管理系统 Esper 中实现 VFDT 算法的研究[7]，而其他数据流算法在 DSMS 中实现的研究尚未见到报道，本文将阐述在 DSMS 中如何实现数据流聚类算法。

最近几年涌现出许多适用于不同数据流应用的 DSMS，文献[8～11]对现有的 DSMS 作了比较详尽的介绍，本文将用到 EsperTech 公司的 Esper(http://esper.codehaus.org)系统。

Esper 系统是一个轻量级、可嵌入的开源软件，而且它是一个基于 Java 开发的软件，因此可以进行功能方面的扩展，如嵌入算法来增加其功能等。Esper 包含了事件流处理(event stream processing，ESP)和复杂事件处理(complex event processing，CEP)引擎，CEP 是一种实时事件处理并从大量事件数据流中挖掘复杂模式的技术。

Esper 提供了事件模式语言去指定基于表达式的模式匹配，同时还提供了事件流的查询语句 EPL(Esper processing language)。EPL 是在 Esper 中用来把数据链匹配需求注册到引擎中去运行的语言，其语法与 SQL 的语法相类似。当 EPL 匹配的数据流流入时，其中一个基础的 POJO 作为接口类构成引擎。EPL 提供了一系列基础的命令，如 create、insert、select、update 以及一些复杂的操作，如命令和合并；此外，Esper 还允许用户快速便捷地建立自定义函数(UDF)来处理数据挖掘算法中的复杂计算。

本文主要介绍在 Esper 中实现基于时间框架的演化数据流聚类算法 Clustream[12] 的在线部分 Micro-clusters 算法。实验证明，在 Esper 系统中集成算法在处理数据量较大的数据流数

收稿日期：2013-07-25；**修回日期**：2013-09-02

作者简介：张晶(1981-)，女，湖南邵阳人，讲师，硕士研究生，主要研究方向为智能信息系统及数据流挖掘算法与数据流管理系统的应用；张阳(1975-)，男(通信作者)，江苏扬州人，教授，博士，博导，主要研究方向为数据挖掘、机器学习等(zhangyang@nwsuaf.edu.cn)。

据时优势比较明显。

1　Micro-clusters 算法简介

本章主要介绍 Clustream 算法[12]中在线处理阶段的 Micro-cluster 算法的基本思想。

定义 1　设数据流样本 $S=\overline{X_1},\overline{X_2},\cdots,\overline{X_k},\cdots$,分别于时间段 $T_1,T_2,\cdots,T_K,\cdots$到达。其中,设每个$\overline{X_i}$都是 d 维的,记做$\overline{X_i}=(x_{i1},x_{i2},\cdots,x_{id})$。每个点$\overline{X_i}$到另一个点$\overline{X_j}$之间的距离定义为

$$\mathrm{dist}(\overline{X_i},\overline{X_j})=\sqrt{\sum_{k=1}^{d}(x_{ik}-x_{jk})^2} \tag{1}$$

每两个相邻时间段之间的时间间隔一定,为 t。设经过初始聚类后,数据一共聚成 q 类。每个数据点到当前已有簇 M_1,$M_2,\cdots,M_q$ 的中心点的距离为 $\mathrm{dist}(\overline{X_i},M_j)(1\leqslant j\leqslant q)$,当任意点$\overline{X_i}$到现有簇中心距离大于设定的最大距离 maxdistance 时,则定义该数据点$\overline{X_i}$为离群点。设任意两个簇 M_i 与 M_j 的中心距离为 clustdist(M_i,M_j),其中,$i\neq j,1\leqslant i\leqslant q,1\leqslant j\leqslant q$,若两个簇中心之间的距离最小,则定义该距离为两簇合并的距离 mergedistance,则有

$$\mathrm{clustdist}(M_i,M_j)=\mathrm{mergedistance} \tag{2}$$

Micro-clusters 算法基本思想如下:

Input:数据流 D;ClusterNum 为当前聚类簇的个数;d 为数据流的维度;t 为处理间隔时间;q 为初始聚类簇的个数;maxdistance 为簇中心到簇内点的最远距离;mergedistance 为合并簇之间的中心距离;initNum 为每批处理的数据个数。

1　初始化参数:ClusterNum = 0,t = 2,d、q、maxdistance、mergedistance、initNum 等参数在运行时从键盘输入;

2　处理第一批进入窗口的 initNum 个数据,利用 K-means 算法聚类;

3　M_jClusterNum = q;//聚成 q 个簇,M_j 为其中一个簇,$1\leqslant i\leqslant q$;

4　for (数据处理未结束){

　for(n = 0;n < initNum;n + +){

　　读取该批数据流中的一个数据点;

　　根据点的位置以及数据块在各维上最大长度,判断是否存在一个包含该点的簇

　　if (exist){

　　修改该簇特征;}

　　else {

　　if dist$P(\overline{X_j},M_i)$ > maxdistance then {

　　为该点创建一个簇;

　　if clustdist(M_i,M_j) < mergedistance then{

　　合并簇 M_i 和 M_j;}//此处保证 ClusterNum = q 不改变

　　}

　　}

　}

2　在 DSMS 中实现流聚类

本章内容将详细介绍在 Esper 系统中实现数据流聚类算法,并实现实时处理大量数据的主要技术与方法。

2.1　在 Esper 中实现数据挖掘算法的技术特点

在 Esper 中实现数据挖掘算法,利用了系统与 Java 语言良好的结合性,虽然 Java 语言编写的原算法并不能直接在系统中应用,而是通过 EPL 对算法进行必要的改写,使之成为 Esper 系统的一个功能性扩展,而此过程中的程序都是利用 Java 语言和 EPL 查询语句来实现的。

在 Esper 系统中实现数据挖掘算法,首先需要进行系统的预备工作,然后建立一个引擎实例,利用 EPL 将算法嵌入实例中的 statement 并生成与之绑定的 listener,实现对数据流的实时处理功能。

2.2　在 Esper 中实现流聚类算法的主要方法

在 Esper 系统中嵌入具有聚类功能的插件,需要用 EPL 及 UDF 来实现。

2.2.1　EPL 的应用

EPL 提供了一系列基础的命令,如 create、insert、select 等,在 Esper 中实现流聚类算法 Micro-clusters 应用这些基础命令完成了大部分的处理过程,如:

```
create window Clusters. win:keepall( ) select maxdistance,initNum,d from Param
create window Clusters(initNum int,maxdistance float,d int)
```

此处利用 create 语句,在系统开始处理数据流之前,为即将被处理的数据流按需求新建一个时间窗口,用于管理及保存其参数设置。

又如:

```
insert into Clusters select maxdistance,initNum,d
```

此处利用 insert 语句,在新的数据流到达系统时,触发事件并将参数数据插入参数窗口中保存,为下一步处理作准备。

```
insert into Clusters initNum = 100.0, maxdistance = 10.0,d = 100
```

此处则利用 insert 语句,对到达时间窗口的数据流的相关参数进行初始化设置,包括设定时间段 t 到达的数据量 initNum、离群点的最大距离 maxdistance,数据点的维度 d 等。

2.2.2　利用 UDF 实现聚类算法

此处应用到了 Esper 中的 UDFs(自定义函数),UDFs 就是为了处理更为复杂、无法单独用 EPL 语言描述的事件或者过程。此处的自定义函数 Clusters$(\overline{X_i},q)$ 里面包含的是 Micro-cluster 算法,利用 Esper 中的事件触发机制和 clusters$(\overline{X_i},q)$ 里面的算法在 Esper 系统中实现数据流的聚类算法。以下详细阐述该过程:

a)利用 select 语句来触发 UDF 中的事件。在此,设置每批数据流到达的时间段 t,单位为 s。例如:

```
select * from ClustersEvent. win:time(t second)
```

若设 t = 2,则表示该事件每 1 s 触发一次,作用就是把到达的数据流按照每 2 s 一批,使之分批进入 Esper 系统中,等待 UDF 中已经改写的 Micro-clusters 算法对其进行聚类。

b)对每一批到达的数据,都需要利用算法找出新到达数据点中的离群点。例如:

```
select * from Clusters where dist > maxdistance
```

找出离群点后,算法会为离群点新建一个簇;在新建簇之前,需要将两个中心距离最近的簇进行合并处理。例如:

```
select * from Clusters where Clustdist = mergedistance
```

c)经过 a)b)两步,系统已经开始对到达的数据流进行聚类,将创建 statement,如:

statement = esperEngine. getEPAdministrator( ) . createEPL(stmt)

d)为创建的 statement 生成一个 listener,并将二者绑定到一起:

statement. addListener(new MicroClusterListener(EventListener))

随着数据流的不断到达,重复执行 b)c),即可在 Esper 中实现 Micro-clusters 算法,达到在 DSMS 中实现数据挖掘功能的目的。

3 实验结果

综上可知,d 维的数据流数据分批进入 Esper 系统中通过算法聚成 q 个簇,期间算法对每个数据只进行一次处理,系统只记录聚类完成后的信息。实验将分别选取 1 维、10 维和 100 维数据分别对在 Esper 系统中实现的算法和直接应用的 Micro-clusters 算法,两种方法分别处理 d 维数据流在数据吞吐量方面进行实验数据对比。

选取 $d=1$ 时的数据来进行实验,实验结果如图 1 所示;选取 $d=10$ 时的数据进行实验,实验结果如图 2 所示;当 $d=100$ 时,实验结果对比如图 3 所示。

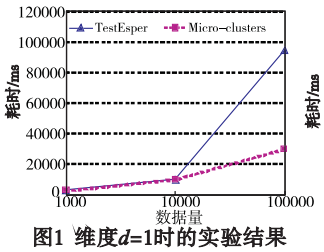

图1 维度d=1时的实验结果

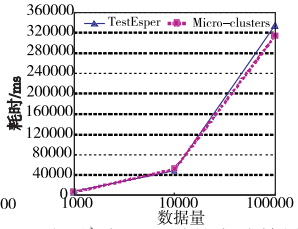

图2 维度d=10时的实验结果

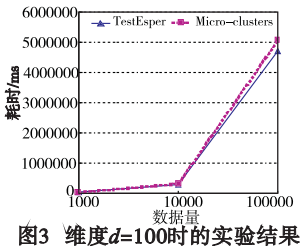

图3 维度d=100时的实验结果

经测试,在测试数据一致时,两种方法得到的聚类结果也是一致的,而在数据量一致的情况下,在 Esper 系统中实现的 Micro-clusters 算法在数据量吞吐及效率上有着与原算法一样的稳定性。

在数据流聚类算法的总耗时中,耗费最大的就是处理流数据的耗时。当数据量较小且维度比较小时,直接用 Micro-clusters 算法的效率明显高很多。因为利用 Esper 系统中集成的算法,需要多花一定的时间先建立时间窗口来处理数据,在此情况下,该过程耗时占总耗时的比例较大,在数据处理过程中时间耗费改变不大,导致时间上没有优势。而随着数据量及维度的增加,原算法是逐个对数据流数据处理的,在此处时间消耗较多;而在 Esper 系统中集成算法处理数据时,虽然仍需花时间建立时间窗口,但在数据处理的过程中,其利用时间窗口定义的数据量一批一批进行,相当于批处理,而且在运行过程中时间窗口可以并行处理,节省了大量的处理数据的时间,使得总耗时比不在 Esper 系统中应用的算法耗时少,因此利用 Esper 系统的优势也逐渐体现出来。

由上可知,当维度和数量级足够大的时候,在 Esper 中应用流聚类算法在稳定性一致的基础上有明显的时间优势。

4 结束语

本文中提出了在 DSMS 中实现 Micro-clusters 算法的方法,该方法是基于大量数据及查询语句检测结果。首先,主要关注如何利用 Esper 系统中用来处理复杂事件或过程的 UDFs 来实现 Micro-clusters 算法的描述,使之转换成为在 Esper 系统可以识别并能直接使用的算法;然后,实现用 EPL 对算法进行调用和事件的触发;最后,分别选取了三种不同维度、不同数据量的数据进行实验。

实验表明,在 Esper 系统中实现流聚类算法是可行而且是具有优势的,在未来的研究中,将研究在 Esper 系统中实现其他数据挖掘功能,以及在 DSMS 中实现数据挖掘功能的方法和机制等。

参考文献:

[1] GAROFALAKIS M, GEHRKE J, RASTOGI R. Querying and mining data streams[C]//Proc of the 2nd ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2002: 635.

[2] 张玲东,毛宇光,曹晨光,等. 数据流管理系统研究与进展[J]. 计算机应用研究,2005,22(6):12-15.

[3] LAW Y N, WANG Hai-xun, ZANIOLO C. Relational languages and data models for continuous queries on sequences and data streams [J]. Association for Computing Machinery Trans on Database Systems, 2011,36(2):8.

[4] THAKKAR H, LAPTEV N, MOUSAVI H, *et al*. SMM: a data stream management system for knowledge discovery[C]//Proc of International Council for Open and Distance Education. 2011:757-768.

[5] SATTLER K, DUNEMANN O. SQL database primitives for decision tree classifiers[C]//Proc of International Conference on Information and Knowledge Management. New York: ACM Press, 2001:379-386.

[6] CALLAGHAN L O , MISHRA N, MEYERSON A, *et al*. Streaming-data algorithms for high-quality clustering[C]//Proc of the 18th International Conference on Data Engineering. 2002: 685-694.

[7] CHAUDHURI S. Data mining and database system: where is the intersection? [J]. Data Engineering Bulletin, 1998,21(1): 4-8.

[8] 袁磊,张阳,李梅,等. 在数据流管理系统中实现快速决策树算法[J]. 计算机科学与探索,2010,4(8):673-682.

[9] MOTWANI R, WIDOM J, ARASU A, *et al*. Query processing, approximation, and resource management in a data stream management system[C]//Proc of the 1st Biennial Conference on Innovative Data Systems Research. 2003.

[10] CHANDRASEKARAN S, COOPER O, DESHANDE A, *et al*. TelegraphCQ: continuous dataflow processing[C]//Proc of ACM SIGMOD International on Management of Data. New York: ACM Press, 2003:668.

[11] CARNEY D, CETINTEMEL U, CHERNIACK M, *et al*, Monitoring streams: a new class of data management applications[C]//Proc of the 28th International Conference on Very Large Data Base. [S. l.]: VLDB Endowment, 2002: 215-226.

[12] AGGARWAL C C, HAN Jia-wei, WANG Jian-yong, *et al*. A framework for clustering evolving data streams[C] //Proc of the 29th International Conference on Very Large Data Base. [S. l.]: VLDB Endowment, 2003:81-92.