一种基于代表点的分布式数据流聚类算法*

高 兵[1,2]，张健沛[1]，杨 静[1]

(1. 哈尔滨工程大学 计算机科学与技术学院，哈尔滨 150001；2. 大连东软信息学院 计算机系，辽宁 大连 116023)

摘 要：为发现分布式数据流下不同形状的聚簇，提出了一种基于代表点的聚类算法。算法首先在代表点定义的基础上，提出环点的概念以及迭代查找密度相连环点的算法，在此基础上生成远程站点的局部模型；然后在协调站点设计合并局部模型，生成全局聚簇的算法。通过真实数据集与仿真数据集的实验表明，算法使用代表点能够发现不同形状的聚簇并显著降低数据传输量，同时通过测试—更新局部模型算法避免了频繁发送数据。

关键词：分布式数据流；数据挖掘；聚类；聚类演化；代表点

中图分类号：TP311 **文献标志码**：A **文章编号**：1001-3695(2012)08-2845-04

doi:10.3969/j.issn.1001-3695.2012.08.011

Representative-based distribute data stream clustering algorithm

GAO Bing[1,2], ZHANG Jian-pei[1], YANG Jing[1]

(1. *College of Computer Science & Technology, Harbin Engineering University, Harbin* 150001, *China*; 2. *Dept. of Computer, Dalian Neusoft Information College, Dalian Liaoning* 116023, *China*)

Abstract: To find the clusters of different shapes under the distributed data streams environment, this paper proposed the representative-based clustering algorithm. First, it presented the concept of circular-point based on the representative points and designed the iterative algorithm to find the density-connected circular-points, then generated the local model at the remote site. Secondly it designed the algorithm to generate global clusters by combining the local models at coordinator site. The experimental results on real and synthetic datasets demonstrate that the algorithm can find the clusters in different shapes and reduce the data transmission by using representative points, while avoiding frequently sending data through the test-update strategy.

Key words: distributed data stream; data mining; clustering; cluster evolving; representative point

0 引言

分布式数据流是指数据分布于多个并发的数据源，每一个数据源产生各自的数据流，且数据随时间改变。例如，在大规模的零售数据仓库中，每一个零售点产生自己的售出项流。研究并开发适合于分布式数据流的高效数据挖掘方法，具有更重要的实际意义。

聚类是数据挖掘领域的一个重要的研究内容，是指对一个已给的数据对象集合，将其中相似的对象划分为一个或多个组(称为簇)的过程。同一个簇中的元素是彼此相似的，而与其他簇中的元素相异[1]。结合以上两方面，如何在分布式数据流中聚类以便发现知识，需要深入研究。

比较直观而简单的处理方法是先将分布的数据流集中于协调站点，再进行聚类。但这种方法因为需要传输全部的数据，存在响应时间长、加重通信负载、不适用于资源受限的环境、隐私保护问题等。

目前的处理方法多采用内网聚集(in-network aggregation)的技术，将处理操作降到网络节点中，计算每一节点的局部汇总，再将汇总信息传输到中心站点，对汇总信息进行二次处理，得到最终的聚类结果。如文献[2]将 K-median 算法应用于分布式流聚类，进行$(1+\varepsilon)$ approximate clustering。文献[3]将 K-center 算法应用于分布式流聚类，这两种都是基于划分的方法，不能处理非球形簇，且需要指定簇数目等划分方法所固有的问题。文献[4]采用了期望最大化(expectation maximization, EM)算法进行分布式数据流聚类，该算法的局限是要求数据符合模型分布，不能很好地处理噪声。

为解决以上问题，本文在基于密度的聚类基础上，提出了一种分布式数据流聚类(representative-based distribute data stream clustering, RB-DDSC)算法。算法包含两个阶段，在远程站点生成局部聚簇模型——代表点集合，在协调站点通过合并算法产生全局聚簇。

本文的主要贡献包括：a)在代表点定义的基础上提出了环点的概念，分析了使用密度相连的环点作为本地模型的合理性，设计了通过迭代查找符合条件的环点来发现本地簇模型的算法；b)设计了分布式数据流环境下的聚类算法，分两个阶段生成基于密度的全局簇模型，能够发现不同形状和不同大小的簇，并且通过测试—更新局部模型算法避免了频繁发送数据，

收稿日期：2011-12-19；**修回日期**：2012-02-09 **基金项目**：国家自然科学基金资助项目(61073043)；黑龙江省自然科学基金资助项目(F201023)

作者简介：高兵(1976-)，男，黑龙江哈尔滨人，讲师，硕士，主要研究方向为数据库、数据挖掘(gaobing@neusoft.edu.cn)；张健沛(1959-)，男，黑龙江哈尔滨人，教授，主要研究方向为数据库、数据挖掘、软件理论；杨静(1962-)，女，黑龙江哈尔滨人，教授，主要研究方向为数据库、数据挖掘、软件理论.

降低数据传输量;c)通过实验表明 RB-DDSC 算法具有良好的聚类质量、较快的处理速度、较少的数据传输量,能够适应分布式流环境下的聚类要求。

1 相关工作

1.1 分布式数据流模型

为了对上文的应用建模,文献[5]首次提出了分布式数据流模型的概念,它是指模型中有若干数据流,来自多个水平分割的数据源,每个点可以观察到自己的数据流,在协调站点,在所有数据流的并集上进行计算,得到最终的统计信息或者进行数据挖掘。

分布式数据流模型的结构如图 1 所示,系统由 M 个远程站点和一个中央协调站点组成。图 1 中的 $S_i(t_s)$ 是远程站点 $site_i$ 时间戳 t_s 上到达的对象集;$Skt_i(t_s)$ 是在时间戳 t_s 上 $site_i$ 向协调站点发送的概要数据结构或由它生成的局部模型,$Skt_i(t_s)$ 因算法不同而采取不同的方式构造。远程站点之间通常也能进行通信,本文仅考虑远程站点只能与协调站点进行双向通信的情形。

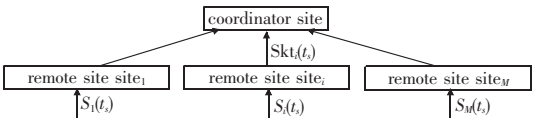

图 1 分布式数据流模型

在分布式数据流模型中,一般不将所有数据全部提交到一个中心位置,而是在分布式计算节点执行一部分靠近数据的计算[6]。当需要时,将得到的局部模型发送到协调站点,在协调站点进行全局计算,得到最终的计算结果。因此,可以得出模型的两个显著优势:

a)全局任务的处理可以由多个远程站点同时进行,提高运算速度。

b)因为发送的是局部模型,可以降低通信负载,避免出现隐私保护等问题。

1.2 概要结构与局部模型

在分布式数据流模型中,由远程站点首先生成的是数据的概要结构,概要结构代表了原有数据的本质特征和基本信息。针对于不同的问题可以使用不同的概要结构,概要结构需要满足以下三个基本性质:

a)压缩。应该比原始数据更小的规模,否则就失去了生成概要结构的意义。

b)合并。集合 P 的概要可以通过计算其划分的概要的汇总得到。即

$$SM(P)=f(SM(P_1),\ SM(P_2))$$

c) 错误界。因为生成的是原始数据的概要信息,概要的解应该保证 $1+\varepsilon$ 近似,ε 是近似因子。

对于一些问题如数据挖掘中的聚类分析,需要对概要结构进行处理得到局部模型,将产生的局部模型发送到协调站点。局部模型需要考虑三个方面的问题:a)能够比较精确地代表原有数据,显然保存的模型信息越多,精确程度越高;b)为了尽可能地减少数据传输,要求局部模型在代表原有数据的基础上,能够尽可能地节省空间;c)能够通过合并或计算局部模型得到全局簇或模式。其中前面两点是两个互相矛盾的问题,在具体问题处理的过程中需要进行权衡。

1.3 密度定义

本文借鉴了 DBSCAN(density-based spatial clustering of applications with noise)算法中的密度定义方法。该算法将具有足够高密度的区域划分为簇,并可以在带有噪声的空间数据库中发现任意形状的聚类。它定义簇为密度相连的点的最大集合,基于密度的聚类的基本想法涉及的定义如下[1]。

定义 1 ε-邻域。一个给定对象周围半径 ε-内的区域称为该对象的 ε-邻域。

定义 2 核心对象。如果一个对象的 ε-邻域内至少包含最小数目 minPts 的对象,那么该对象称为核心对象。

定义 3 直接密度可达。给定一个对象集合 D,如果 p 是在 q 的 ε-邻域内,而 q 是一个核心对象,则对象 p 从对象 q 出发是直接密度可达的。

定义 4 密度可达(density-reachable)。如果存在一个对象链 $p_1,p_2,\cdots,p_n,p_1=q,\ p_n=p$,对 $p_i\in D(1\leqslant i\leqslant n)$,$p_i+1$ 是从 p_i 关于 ε 和 minPts 直接密度可达的,则对象 p 是从对象 q 关于 ε 和 minPts 密度可达的。

定义 5 密度相连(density-connected)。如果对象集合 D 中存在一个对象 o,使得对象 p 和 q 是从 o 关于 ε 和 minPts 密度可达的,那么对象 p 和 q 是关于 ε 和 minPts 密度相连的。

一个基于密度的簇是基于密度可达性的最大的密度相连对象的集合。不包含在任何簇中的对象被认为是噪声。

2 RB-DDSC 算法

本章给出分布式数据流上的聚簇算法 RB-DDSC 的描述。该算法的整体结构包括两个阶段:远程站点处理阶段和协调站点处理阶段,如下所示:

$RB\text{-}DDSC\_Remote_i$

a) generate local model composed of represent points

b) if time changes, maintain local clusters CLU_i

c) $\triangle CLU_i(t_{current})\leftarrow CLU_i(t_{current})-CLU_i(t_{current}-1)$

d) if $\triangle CLU_i(t_{current})>0$ send($\triangle CLU_i(t_{current})$)

RB-DDSC_Coord

a) $GLO_0\leftarrow\cup_i\triangle CLU_i(t_{current})$

b) maintain global clusters

每一个远程站点产生自身的概要结构,称为代表点,由代表点的集合构成局部聚簇模型,将局部聚簇模型发送到协调站点,当有新的数据点到来时,更新模型并发送更新。协调站点收到各个远程站点发送的局部聚簇模型后,对得到的局部模型进行再次计算,产生全局聚簇模型,同时当收到更新的局部聚簇模型后,更新全局聚簇模型。

3 远程站点处理

3.1 代表点的定义

在远程站点通过 DBSCAN 算法可以产生数据集合的核心点,由于这些核心点代表了 ε-邻域内的 minPts 个点 ,它们的汇总可以作为远程站点产生的局部模型,被发送到协调站点进行全局聚类。但是,核心点的数量可能是非常大的,在聚簇的密集区域更是如此,简单地发送核心点集合将导致过多的传输量及过长的传输时间,所以直接采用 DBSCAN 算法生成并传输核心点不适用于分布式数据流的情况。

因为核心点可能是密集的,不需要传输所有的核心点,可以在核心点中选择一个子集来代表全部的核心点,由此给出代表点的定义。

定义 6　代表点[7]。设 C 是数据集中关于 ε 和 minPts 得到的一个聚簇。$Cor_C \subseteq C$ 是属于聚簇 C 中的核心点集合,称满足如下三个条件的集合 $Rcor_C \subseteq C$ 为代表点集合:

$$Rcor_C \subseteq Cor_C \tag{1}$$

$$\forall s_i, s_j \in Rcor_C : s_i \neq s_j \Rightarrow s_i \notin N_\varepsilon(s_j) \tag{2}$$

$$\forall c \in Cor_C, \exists s \in Rcor_C : c \in N_\varepsilon(s) \tag{3}$$

如图 2 所示,A、B、C 都是聚簇 C 中的核心点,如果 A 是代表点集合 $Rcor_C$ 中的一个元素,因为点 B 在 A 的 ε-邻域内,所以 B 不属于 $Rcor_C$。因为 C 不在 A 的 ε-邻域内,所以 C 可以属于 $Rcor_C$。相反,如果 B 是代表点集合 $Rcor_C$ 中的一个元素,因为点 A 与 C 都在 B 的 ε-邻域内,所以它们都不属于 $Rcor_C$。因此可能存在几种不同的 $Rcor_C$ 集合都满足定义 6。具体的代表点集合由数据的实际处理顺序决定。

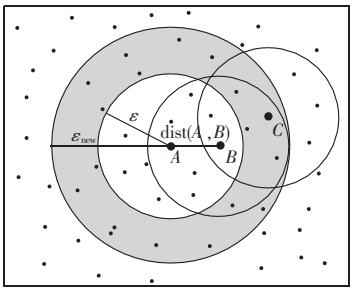

图 2　局部模型(代表点及 ε_{new})

3.2　局部模型

假设在一个远程站点 k 找到 n 个聚簇 $C_1, \cdots, C_n$,在局部模型中,用代表点的集合 $Rcor_{C_i}$ 来代表每一个局部聚簇。局部模型就是由不同 $Rcor_{C_i}$ 集的并集构成的。

使用基于密度的聚类方法,经常会有若干个核心点位于另一个核心点的 ε-邻域内,当使用一个较大的 ε 值时更是如此。如图 2 所示,假设选择核心点 A 为代表点,则 $A \in Rcor$ 且 $B \notin Rcor$,因为 B 不满足定义 6 中的第二个条件。在这种情况下,A 作为代表点不仅应该代表 A 自身的 ε-邻域内的数据,也应该同时代表 B 的 ε-邻域内的数据,所以 A 代表数据集 $N_\varepsilon(A) \cup N_\varepsilon(B)$,必须为 A 重新赋予一个新的邻域范围值 $\varepsilon_A = \varepsilon + \mathrm{dist}(A,B)$。其中 $\mathrm{dist}(A,B) < \varepsilon$ 是核心点 A 与 B 的距离。

必须为所有的代表点赋予一个新的 ε,用来指出代表点所能代表的数据的范围,如果 $Rcor \subseteq C$ 是代表点的集合,对于每一个 $S \in Rcor$,指定新的 ε_s 表示 S 所代表的区域,即

$$\varepsilon_s = \varepsilon + \max\{\mathrm{dist}(s, s_i) \mid s_i \in Cor \wedge s_i \in N_\varepsilon(s)\} \tag{4}$$

性质 1　ε_s 的最大值小于等于 2ε,这是代表点所代表区域的最大范围,即 $\varepsilon_s \leqslant 2\varepsilon$。

证明　通过式(3)(4)易知。

定义 7　环点。位于 ε 和 ε_{new} 之间的点称做环点。例如图 2 中位于灰色部分的点 C。

性质 2　在同一个簇中的代表点一定是环点。

证明　根据定义 6,如果点 a 是代表点,那么位于 $N_\varepsilon(a)$ 中的点一定不是代表点;环点满足式(2)的条件,又因为在同一个簇中,代表点的范围一定是相连的,所以结论成立。

本文的算法通过迭代地查找环点,得到由代表点构成的局部模型,如下所示。每一个代表点和得到的对应 ε_s 值以元组的形式保存,由所有元组的集合构成了站点 k 的局部模型。局部模型被发送到协调站点进行全局聚簇,同时保存噪声数据以便进行后续的更新处理。

局部模型算法如下:

```
localModel(data)
    for every point p do
    if p.unClassified{
    if expandCluster(p) clusterID++;}
expandCluster(p)
    if p is representative {
    save(p,εq);
    getListofCircularPoints(p);
    }
    for every point q in listofCircularPoints(p){
    if q is core point{
    set q as representative;
    save(q,εq);
    theListofCircular = addListofCircularPoints(q);
    }
    else { set q as noise; remove q;}}
```

性质 3　设在远程站点 j 已经得到 n 个簇 $C_1, \cdots, C_n$,对于每一个簇 C_i,用 $|Rcor_{C_i}|$ 表示其中的代表点数,则每一个远程站点发送到协调站点的全部的代表点数 m 为

$$m_{\mathrm{remote}_j} = \sum_{i=1,\cdots,n} |Rcor_{C_i}|$$

3.3　测试—更新算法

在远程站点中保存由代表点构成的局部模型和噪声点,当有新的数据点到来时,首先测试该点是否包含于局部模型,如果该点属于局部模型的范围,保持局部模型不变。如果该点不在局部模型的范围之内,需要判断这个点是否可以和噪声点合并,生成新的聚簇,或者将这个点单独保存为噪声点。如果是新生成的簇,需要判断这个簇是否独立,或者是否和已有模型合并,这两种情况都会改变局部模型,需要更新局部模型并发送到协调站点。

测试—更新算法如下:

```
TestAndUpdate(localModel,data)
  for each data do
  if (fitModel(localMode,data))
      return;
  else{
  if(isNoise(data))
    save data;
  else
    updateModel(localMode,data);
    send(localMode);
  }
```

根据该算法,新到的数据点使用当前的局部模型进行测试,针对于数据流中数据不断到达的特点,该方法具有较好的响应时间。先测试数据点是否属于现有模型,如果属于现有模型则不需要更新,当数据模型比较稳定时,避免了频繁更新模型和频繁发送数据。

4　协调站点处理

局部代表点是由一对 (r, ε_r) 构成,代表了包含于 r 的 ε_r 邻域 $N_{\varepsilon_r}(r)$ 中的所有数据点,所有包含于 $N_{\varepsilon_r}(r)$ 中的数据属于同一个聚簇。因此,每一个局部代表点都形成了一个自身的簇,在协调站点需要检查来自于不同远程站点中的代表点是否可以进行合并,生成全局聚簇。最终的全局模型由合并后的代

表点和未合并的单独代表点所构成,如图 3 所示。

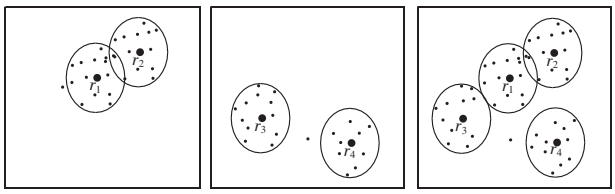

图 3 全局模型

性质 4 设共有 m 个站点,根据性质 3,在协调站点接收到的代表点总数,可以通过下式计算:

$$\text{num} = \sum_{j=1,\cdots,m} m_{\text{remote}_j} = \sum_{j=1,\cdots,m} \sum_{i=1,\cdots,n} |Rcor_{C_i}|$$

因为所有的代表点已经是其自身簇的表示,所以当合并簇时,将邻域内的代表点个数设置为 2,这时本文使用的合并算法可以简化问题的处理,当两个代表点间的距离小于两个代表点邻域半径之和时,即 $\text{dist}(a,b) < \varepsilon_a + \varepsilon_b$ 时进行合并,全局聚簇算法如下所示:

```
globalModel( Rcor )
  select represent from cluster at every sites
  for each represent r1 do
    compute dist( r1 , r2 )/ * r2 come from another site * /
    if( dist( r1 , r2 ) < εr1 + εr2 ) {
      merge ( r1 , r2 )
  }
```

当有局部模型的改变被传送到协调站点时,产生聚簇更新问题,通过上文对局部模型更新的分析可知,在协调站点只需要检查改变后的模型对现有模型的影响即可。如果与现有模型不相交,产生一个新的聚簇,如果与现有模型相交,进行聚簇合并。

5 实验评估

为了评估 RB-DDSC 算法的聚类效果,本文在数据挖掘软件 Weka 3.6.0 环境中,分别实现了 RB-DDSC 算法和修改后的分布式 DBSCAN 算法。所有的实验在 P2 CPU(1 GHz)、1 GB 内存环境下运行,实验分别使用了仿真数据集和 UCI Letter 数据集。

本文使用三个二维仿真数据集评估算法的质量,其中的两个数据集来自于 DBSCAN 算法测试用的数据集 Db1 和 Db2[8],最后一个数据集 Db3 来自于参考文献[9]。实验结果在表 1 中给出,可以看出代表点能够发现正确的簇个数,能够发现三个数据集中任意形状的簇。在数据集 Db3 中算法得到的簇数少于实际的数目,这是因为代表点使用了调整后更大的邻域值 ε_{new},使两个邻近的簇被合并。

表 1 算法质量比较

比较项	Db1	Db2	Db3
cluster number	4	4	25
RB-DDSC	4	4	23

本文使用 UCI Letter 数据集评估方法的效率和传输量,数据集中包含 20 000 个 16 维的数据。实验结果在图 4、5 中给出。从图 4 可以看出,随着站点数的增加,算法的运行时间显著降低,这是因为代表点的计算可以并发执行。图 5 显示了随着数据量的增加,算法产生的代表点数远少于由分布式的 DBSCAN 算法产生的核心点数,由此极大地降低了算法的传输量,这是由前文的分析所得的。

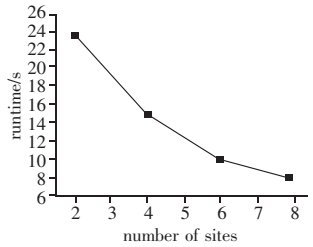

图 4 算法运行效率

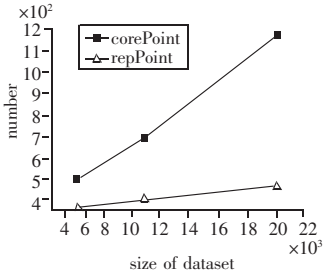

图 5 传输量 DBSCAN vs DB-DDSC (MinPts:20)

6 结束语

聚类是分布式数据流中知识发现的重要内容,如何在分布的流数据的情况下发现聚簇信息有着重要意义。本文提出了一种新的基于代表点的两阶段聚类算法 RB-DDSC,在远程站点用代表点的集合表示局部簇模型,在协调站点用基于距离的算法生成全局聚簇模型。通过分析和实验表明,该方法能够得到较好的聚类效果。

后续的工作将对算法作进一步的实验比较,重点比较聚类的质量和噪声点的识别处理;在此基础上,进一步改进算法以适应滑动窗口模型下的聚簇。

参考文献:

[1] HAN Jia-wei, KAMBER M. Data mining: concepts and techniques [M]. 2nd ed. San Fransisco: Morgan Kaufmann, 2006: 467-589.

[2] ZHANG Qi, LIU Jin-ze, WANG Wei. Approximate clustering on distributed data streams[C]//Proc of the 24th IEEE International Conference on Data Engineering. 2008: 1131-1139.

[3] HUANG Jiang-hua, ZHANG Jun-ying. Fuzzy C-means clustering algorithm with spatial constraints for distributed WSN data stream[J]. International Journal of Advancements in Computing Technology, 2011, 3(2): 165-175.

[4] ZHOU Ao-ying, CAO Feng. Distributed data stream clustering: a fast EM-based approach[C]//Proc of the 23rd International Conference on Data Engineering. 2007.

[5] GIBBONS P, TIRTHAPURA S. Estimating simple functions on the union of data streams[C]//Proc of ACM Symposium on Parallel Algorithms and Architectures. 2001: 281-291.

[6] HUANG Jiang-hua, ZHANG Jun-ying. Distributed dual cluster algorithm based on grid for sensor streams[J]. Journal of Digital Content Technology and Its Applications, 2010, 4(9): 225-233.

[7] JANUZAJ E, KRIEGE HP, PFEIFLE M. Towards effect and efficient distributed clustering[C]//Proc of the 3rd IEEE International Conference on Data Mining. 2003.

[8] ESTERM M, KRIEGE H P, SANDER J, *et al*. A density-based algorithm for discovering clusters in large spatial databases with noise [C]//Proc of the 2nd International Conference on Knowledge Discovering in Databases and Data Mining. Massachusetts: AAAI Press, 1996: 226-232.

[9] ZHOU Shui-geng, ZHOU Ao-ying, JIN Wen, *et al*. FDBSCAN: a fast DBSCAN algorithm[J]. Journal of Software, 2000, 11(6): 735-744.