基于数据概要描述的分布式数据流聚类模型与算法

毛国君[1] 曹永存[2]
(中央财经大学信息学院 北京 100081)[1] (中央民族大学信息工程学院 北京 100081)[2]

摘 要 数据流挖掘可有效解决大容量流式数据的知识发现问题,并已得到广泛研究。数据流的一个典型的例子是传感器采集的流式数据。然而,随着传感器网络的应用普及,这些流式数据在很多情况下是分布式采集和管理的,这就必然导致分布式地挖掘数据流的需求。分布式数据流挖掘的最大障碍是由分布式而导致的挖掘质量或者效率问题。为适应分布式数据流的聚类挖掘,探讨了分布式数据流的挖掘模型,并且基于该模型设计了对应的概要数据结构和关键的挖掘算法,给出了算法的理论评估或者实验验证。实验说明,提出的模型和算法可以有效地减少数据通信代价,并且能保证较高的全局模式的聚类质量。

关键词 分布式数据流,数据概要,增量式聚类,全局模式
中图法分类号 TP311 **文献标识码** A

Clustering Models and Algorithms for Distributed Data Streams Based on Data Synopsis

MAO Guo-jun[1] CAO Yong-cun[2]
(School of Information, Central University of Finance & Economics, Beijing 100081, China)[1]
(School of Information Engineering, Minzu University of China, Beijing 100081, China)[2]

Abstract Mining data streams aims at discovering knowledge from a large of streaming data, in which enough efforts have been done in recent years. As a typical example, the data to be collected by a sensor is a format of data streams. However, in the technical environment of a sensor network, multiple sensors always are set and they collect data in a distributed way, so mining data streams with a distributed way is making a challenge issue. Most ongoing studies for mining distributed data streams are suffering from the problems of accuracy or efficiency. In this paper, the model for clustering a distributed data stream was discussed, including a new synopsis data structure for summarizing data streams and some effective algorithms for key mining phases. The reasons of presented algorithms were also discussed. Experimental results demonstrate that presented models and algorithms have less transmission cost and higher clustering quality to mine the global pattern from distributed data streams.

Keywords Distributed data stream, Data synopsis, Incremental clustering, Global pattern

1 引言

数据流(data stream)作为一种新型的数据组织与处理形式,已经得到广泛研究[1]。简言之,数据流是一种随时间变化、动态增长的数据序列。传感器采集的数据具有典型的数据流特征。然而,随着传感器网络的应用深入,不仅传感器网络的每个传感器采集的数据可以看作是随时间增长的数据流,而且整个传感器网络聚集的数据形成一个多节点分布的相互关联的多数据流,这类应用可以借助于近年提出的分布式数据流(distributed data stream)概念来加以探索。从分布式数据流挖掘角度看,每个传感器形成的局部数据流隐含着局部的模式,反映局部数据的变化情况,但是整个传感器网络形成的多数据流及其关联性的挖掘可能发现全局性的知识模式。发现全局性的知识模式已经成为分布式数据流挖掘的核心问题之一。

分布式数据流的挖掘近年受到关注[2]。当前挖掘全局模式有两种基本策略:(1)数据集成,即首先将所有局部节点数据传给中心节点,再在中心节点进行全局模式挖掘;(2)模式集成,即首先在各个局部节点进行局部模式挖掘,再将局部模式上传给中心节点进行全局模式集成。这两种策略都有着明显的优缺点。前者虽能保证数据的完整性,从而获得高精度的全局模式,但是数据传输代价太大,难以满足实际应用的需求。后者虽可以使传输量大幅下降,但模式集成难度大,缺乏指导性,很难保证全局模式挖掘的质量和精度。因此,分布式数据流挖掘的目标应该是:寻找高效的模型,使得节点间通讯代价最小,而分布式挖掘的结果可以和集中式的挖掘效果相媲美。从这个意义上说,研究分布式数据流的模式集成技术将面临着效率和精度的双重挑战,从效率上应该更注重通讯代价的控制;从挖掘精度上应该着重解决由于分布挖掘导致的全局模式形成的误差控制。

到稿日期:2012-09-12 返修日期:2012-12-27 本文受国家自然科学基金项目(62173293),中央财经大学教改项目基金资助。
毛国君(1966-),男,博士,教授,主要研究方向为数据挖掘,E-mail:maximmao@hotmail.com;曹永存(1962-),男,教授,主要研究方向为数据库和数据挖掘。

利用数据概要(data synopsis)来解决分布式数据流挖掘问题可以在数据传输代价和挖掘精度上获得平衡[3-6]。所谓数据概要,就是一个数据流的数据分布对应的统计值的集合。在文献[3]中,作者给出的概要数据结构为:〈簇内的数据和、平方和、数据个数〉;利用这样的概要数据结构,作者进而提出了一种分布式等宽聚类算法(Distributed Fixed-width Clustering Algorithm,DFCA),并解决了分布式传感器网络的异常检测问题。整体上看,目前研究的数据概要主要还是包含基本的数据统计信息(如平方和、均值、均方差等)[1,3,4],直接利用它们很难获得精确的全局模式。特别是,现有的方法缺少在全局模式演变过程中评价挖掘质量的机制,或者说缺少对挖掘过程的启发性指导,使得全局模式的生成存在盲目性。

本文设计了一种改进的数据流的概要数据结构,通过增加簇内凝聚度等指标来加强对簇质量的评估。同时,设计了一个新颖的还原样本点算法,该算法可以将数据概要的局部统计值转变成全局学习的样本集合。这样,全局模式就可以通过还原的样本集加以学习。基于这样的思想,本文给出了分布式数据流的全局模式挖掘对应的几个核心算法,并详细讨论了相关问题。

2 问题描述

定义 1(数据流) 一个数据流定义为一个时间序列上的数据元组集合。即给定时间序列 $time\_series=\langle 1,2,\cdots,t,\cdots\rangle$,它上的一个数据流表示为 $data\_stream=\langle tuple_1,tuple_2,\cdots,tuple_t,\cdots\rangle$,其中 $tuple_t$ 对应 t 时刻的数据元组值。

定义 2(分布式数据流) 一个分布式数据流定义为若干个在同一个时间序列上的(单)数据流的集合。即给定一个时间序列 $time\_series$,基于该时间序列之上的分布式数据流可以描述为一个集合,即 $distributed\_data\_stream=\{data\_stream^1,data\_stream^2,\cdots,data\_stream^n\}$,其中 $data\_stream^k$ $(k=1,2,\cdots,n)$是定义 1 所定义的数据流,n 为节点个数。

分布式数据流有同构和异构之分。假如一个分布式数据流的所有(局部)数据流都是建立在同一个元组结构描述基础上的,那么它被称为是同构的[7]。如果一个分布式数据流的局部数据流存在不一致的元组结构描述,那么它就被称为是异构的[8,9]。例如,在一个传感器网络中,所有的节点如果都收集同样的监测指标(集)来形成数据流,那么就构成同构的分布式数据流;假如不同的监测节点监测不同的指标(集),如有的监测温度,有的监测湿度等,那么就形成异构的分布式数据流。此外,时间同步也是一个重要问题。受设备、传输等因素影响,不同节点可能在数据采集上存在所谓的时间相位问题。如果所有的数据流不存在或者可以忽略时间相位问题,那么这样的分布式数据流就被认为是时间同步的。本文只考虑同步同构的分布式数据流。

定义 3(时间数据窗口) 设时间序列 $time\_series$ 及其之上的一个同步同构的分布式数据流 $DDS=\{DS^1,DS^2,\cdots,DS^n\}$。给定 $time\_series$ 的一个有限的时间区间 $T=(i,j]$ $(i<j)$,DDS 在 T 上的一个时间数据窗口定义为:$Window(T,DDS)=\{D_T^1,D_T^2,\cdots,D_T^n\}$,其中 $D_T^k(k=1,2,\cdots,n)$是(单)数据流 $DS^k(k=1,2,\cdots,n)$在 T 时间段的数据集。

由于数据流是潜在无限的,因此窗口是利用有限的内存空间解决无限数据流问题的基本手段。对一个时间序列 $\langle 1,2,\cdots,t,\cdots\rangle$,设被观察的数据流在 $(t-1,t]$上的时间数据窗口为 W_t,假如已经挖掘出 t 时刻的知识模式 P_t,那么基于滑动窗口技术的数据流上的增量学习就是解决利用 $\langle P_t,W_{t+1}\rangle$来形成 P_{t+1}的问题。如前所述,为了提高分布式数据流全局模式挖掘的精度,将在局部节点生成和保存数据概要。这样,分布式数据流节点级处理就需要考虑增量式的模式挖掘和数据概要更新两种方法。

定义 4(节点级数据流挖掘模型) 设时间序列为$\langle 1,\cdots,t-1,t,\cdots\rangle$,节点级的数据流为 D。给定被观察的时间点 t,则 D 在时刻 t 的挖掘模型可以表示为:

$\langle W_t,pattern_{t-1},synopsis_{t-1};$
$local\_mining\_method,summarizing\_method;$
$pattern_t,synopsis_t\rangle$

其中,$local\_mining\_method$ 和 $summarizing\_method$ 分别是知识挖掘和概要生成算法。它们利用数据流的当前窗口数据和上一个窗口得到的模式和概要,增量式地形成当前时刻的模式和概要。

定义 5(分布式数据流挖掘模型) 设时间序列$\langle 1,\cdots,t-1,t,t+1,\cdots\rangle$和在该时间序列之上的一个分布式数据流为 $DDS=\{DS^1,DS^2,\cdots,DS^n\}$。给定被观察的时间点 t,则 DDS 在时刻 t 的全局挖掘模型可以表示为:

$\langle p_t^1,p_t^2,\cdots,p_t^n,s_t^1,s_t^2,\cdots,s_t^n,$
$global\_pattern_{t-1},global\_synopsis_{t-1};$
$global\_mining\_method,summarizing\_method;$
$global\_pattern_t,global\_synopsis_t\rangle$

其中,p_t^k 和 $s_t^k(k=1,2,\cdots,n)$ 是局部数据流 D^k 在 t 时刻的局部模式和概要,$global_pattern_{t-1}$和 $global_synopsis_{t-1}$是基于 $t-1$ 时刻中心节点获得的全局模式和全局概要,它们构成 t 时刻全局学习的输入型数据;通过 global_mining_method、summarizing_method 方法可以在中心节点挖掘出 t 时刻的全局模式和全局概要。

定义 6(簇凝聚度) 给定一个簇 c,假设它的中心点也记为 c,则它的凝聚度(Cohesion)可以定义为:

$$\text{Cohesion}(c)=\sum_{i=1}^{n}d(p_i,c)^2$$

式中,n 为簇 c 所包含的样本点的个数,$d(p_i,c)$表示簇 c 内的样本点 $p_i(1\leqslant i\leqslant n)$与该簇中心点 c 之间的距离。

簇凝聚度反映了一个簇内的数据的耦合程度,一个好的簇必须保证簇内数据耦合程度要高。按照定义 6,一个簇的凝聚度的值越小,表明它的耦合度越高。同时,本文也把簇凝聚度作为数据概要的重要元素之一。

3 分布式数据流的聚类模型和算法

本文解决问题的基本思路是:以时间窗口为单元,在局部节点进行聚类形成 t 时刻的簇集和数据概要;将局部节点产生的数据概要传送到中心节点;在中心节点根据传来的 t 时刻的所有数据概要,以局部簇为单位还原符合它的数据概要的学习样本点,集成这些样本点并利用它们形成 t 时刻的全局聚类结果。模型 1 给出了对应的描述。

模型 1 分布式数据流全局聚类模型 DGCM(Distributed Global Clustering Model)

已知:

时间序列$\langle 1,2,\cdots,t,\cdots\rangle$上一个分布式数据流。

求解：

t 时刻全局聚类结果。

方法：

模块 1（数据收集）：n 个局部节点同时收集所在节点的数据流，按照时间数据窗口形成当前数据集。

模块 2（局部挖掘）：在局部节点形成 t 时刻的局部簇集和数据概要，本文使用增量式局部最远点聚类算法 IFP（后面介绍）。

模块 3（信息传送）：将局部节点的 t 时刻的簇集对应的数据概要传送到中心节点。

模块 4（全局挖掘）：使用 t 时刻得到的局部数据概要信息，生成全局学习样本并使用它们进行聚类，得到全局聚类结果。本文使用还原样本点算法 RSP 进行样本数据生成（后面介绍），并使用全局凝聚聚类算法 GAC 来形成全局聚类结果（后面介绍）。

附注：除非需要必要的等待，在连续时刻 t、t+1、t+2 和 t+3，模块 1－模块 4 可以并行操作。

忽略掉具体的技术细节，DGCM 模型主要是解决模块 2 和模块 4 对应的核心算法设计问题。因此，本文的剩余部分主要聚焦在这些算法的设计问题上。

3.1 局部增量式聚类算法

聚类算法近年已得到很好的研究与发展，已出现了很多诸如 k-means[6,11]、k-medoids[9,11]、k-furthest-Points[3,4,11] 等经典算法的改进版本，使之能够适应数据流聚类挖掘的各种不同的应用场景。本文在处理局部节点模式挖掘时，对 k-furthest-Points 方法进行改进。主要改进在两个方面：(1)为适应数据流的窗口处理机制，在每个窗口数据测试后进行最优的中心点集合测试，尝试找出最远距离的中心点集合。这样，在窗口改变中实现增量式的中心点改变。(2)增加聚类后整体的簇凝聚度的测试来保证聚类的质量。算法 1 描述了详细的处理过程。

算法 1 增量式局部最远点聚类算法 IFP(Incremental Furthest Point)

输入：簇数目上限 k(k≥2)；

t－1 时刻的簇中心集合 $P=\{p_1,p_2,\cdots,p_k\}$；

t－1 时刻的簇集 $C=\{c_1,c_2,\cdots,c_{k-1},c_k\}$；

簇中心之间最小距离 MinL；

t 时刻的窗口数据 w_t；

O 为潜在离群点集合。

输出：对应 t 时刻，更新的 C，P 和 MinL。

方法：

(1) FOR each $p\in W_t$；

(2) Find its closest cluster c_i in C, i. e. $d(p,p_i)=\min\{d(p,p_j)\mid j=1,\cdots,k\}$；$p_i$ 是 c_i 簇中心

(3) IF $d(p,p_i)\leqslant$MinL THEN Put p into cluster c_i；

(4) ELSE Put p into O；

(5) ENDFOR；

(6) IF O is not NULL

(7) THEN Clustering O into k clusters with the centers Q；

(8) Select the k furthest points from P and Q, recluster C and O into k clusters, called C*；

(9) Calculate $h=\sum_{c\in C}\text{Cohesion}(c)$ and $h^*=\sum_{c^*\in C^*}\text{Cohesion}(c^*)$

(10) IF $h^*<h$ THEN use C* to update C and its related P and MinL.

通过算法 1 可以看出：在选取重聚类的中心点时，首先使用距离最远的中心点集合进行尝试，然后把它和原来的簇集的整体簇凝聚度（所有簇的凝聚度之和）进行比较，只有当整体簇凝聚度的值下降时才进行替换，否则仍然维持原来的簇集合。这样做主要是考虑：中心点最远虽然可以保证簇之间的耦合程度很低，但是有可能使某些簇内的耦合程度下降。由于本文引入了簇凝聚度，我们可以在尝试改变聚类结果后通过衡量整体聚类质量来决定是否采用新的聚类结果。另外，在进行一个窗口的聚类中，并不是在处理每个新的点处理时就尝试改变簇中心，而是在所有窗口数据处理完再尝试改变簇中心来测试聚类效果，这主要是为了提高效率。

3.2 局部概要数据结构

数据概要一般是由原始数据的统计值构成，本文的概要数据结构如表 1 所列。

表 1 概要数据结构

属性名称	含义
Centre	簇的中心
Radius	簇的半径
Dimension	数据的维度数
StandardDeviation	簇内数据的标准方差
SampleNumber	簇的数据点数目
Cohesion	簇的簇凝聚度

3.3 还原样本点算法

从分布式挖掘构架上，局部节点的聚类已经产生了局部模式，但是全局模式不是局部模式的简单堆积，需要重新生成。对于聚类而言，假如有 n 个局部节点，每个节点生成 k 个簇，那么就需要把这 $n\times k$ 个簇生成适量大小的全局簇。还原点样本算法就是根据得到的数据概要集合在中心节点处形成新的样本数据集合。利用这些样本数据就可以重新聚类来得到全局意义的聚类结果。由于本文基于数据概要结构进行局部的数据归纳和传送，因此还原点样本算法产生的样本数据集合必须符合数据概要所对应的统计值信息。算法 2 给出了还原点样本算法的具体过程，定理 1 则说明了这种还原样本点的合理性。

算法 2 还原样本点算法 RSP(Revert Sample Points)

输入：s_t^i：局部节点 i 在 t 时刻产生的一个数据概要

输出：r_t^i：局部节点 i 在 t 时刻产生的全局学习样本集

方法：

(1) $\mu\leftarrow s_t^i$. Centre；

(2) count $\leftarrow s_t^i$. SampleNumber；

(3) r $\leftarrow s_t^i$. Radius；

(4) d $\leftarrow s_t^i$. Dimension；

(5) FOR i=1 to count

(6) FOR j =1 to d

(7) $rand_j$=Rand()；利用随机函数产生一个随机数

(8) $p_j\leftarrow\mu_j+r*rand_j$；//$\mu_j$ 是 μ 第 j 维值

(7) ENDFOR

(8) $p\leftarrow\langle p_1,p_2,\cdots,p_d\rangle$；//合成数据点 p

(9) Add p into r_t^i；

(10) ENDFOR

定理 1(RSP 算法维持基本的统计信息不变) 设一个簇内的原始数据集为 P，且服从均值为 μ 和方差为 σ^2 的正态概率分布；对该簇使用算法 RSP 还原的数据样本集为 R；若对于任一指定的维度 X，RSP 在该维度上使用的随机函数 *Rand*()产生满足均值 μ_x 和方差 σ_x^2 的正态概率分布的[－1,1]的随机数。则：(1)R 是有界的；(2)R 仍然服从均值为 μ 和方差为 σ^2 的正态概率分布。

证明：以二维数据为例，按照定理假设，P 的均值表示为 $\mu=\langle\mu_x,\mu_y\rangle$，标准差是 $\sigma^2=\langle\sigma_x,\sigma_y\rangle$；RSP 算法还原的样本点集合 R 可以表示为 $\{(x_i,y_i)\mid i=1,2,\cdots,n\}$。则：

(1)根据簇(圆形)的几何意义，很容易得到：P 的最理想的中心点是 (μ_x,μ_y)，且 X 维的数值在 $[\mu_x-r,\mu_x+r]$ 附近。对于使用 RSP 还原的任意数据样本点 (x_i,y_i)，根据 RSP 算法步骤(6)：$x_i=\mu_x+r\times rand_x$。因为 $rand_x$ 是通过 $Rand()$ 函数产生的 $[-1,1]$ 中的数，所以 x_i 必然落在 $[\mu_x-r,\mu_x+r]$ 中。Y 维的数值同理可证。所以 RSP 算法还原的数据点和原始数据点的上下界基本相同。

(2)对于 R 上的所有样本点 $\{(x_i,y_i)\mid i=1,2,\cdots,n\}$，计算 X 维的均值和方差 A_x 和 S_x，把它们和原始的数据集 P 在 X 维的 μ_x 和方差 σ_x^2 进行比较：

$$A_x=\frac{x_1+x_2+\cdots+x_n}{n}=\frac{\mu_x+r\times rand_1+\cdots+\mu_x+r\times rand_n}{n}=\frac{\mu_x\times n+r\times\sum_{i=1}^{n}rand_i}{n} \tag{1}$$

因为 $(rand_1,rand_2,\cdots,rand_n)\sim N(0,\frac{\sigma_x^2}{r^2})$

所以 $\frac{\sum_{i=1}^{n}rand_i}{n}=0$

故(1)$=\frac{\mu_x\times n}{n}+r\times\frac{\sum_{i=1}^{n}rand_i}{n}=\mu_x$。

$$S_x=\frac{\sum_{i=1}^{n}(x_i-\mu_x)^2}{n}=\frac{\sum_{i=1}^{n}(\mu_x+r\times rand_i-\mu_x)^2}{n}=\frac{\sum_{i=1}^{n}r^2rand_i^2}{n}=\frac{r^2\sum_{i=1}^{n}rand_i^2}{n} \tag{2}$$

因为 $(rand_1,rand_2,\cdots,rand_n)\sim N(0,\frac{\sigma_x^2}{r^2})$

所以 $\frac{\sum_{i=1}^{n}(rand_i-0)^2}{n}=\frac{\sum_{i=1}^{n}rand_i^2}{n}=\frac{\sigma_x^2}{r^2}$。

故(2)$=r^2\times\frac{\sum_{i=1}^{n}rand_i^2}{n}=r^2\times\frac{\sigma_x^2}{r^2}=\sigma_x^2$。

同理，Y 维可证。

大于二维的数据证明过程类似。证毕。

3.4 全局聚类算法

在聚类算法中，划分和层次方法是最为常见的两种聚类技术，其中划分方法具有较高执行效率，而层次方法则更多考虑数据特性，其聚类效果较好。相对于局部节点而言，中心节点不必过分追求聚类时效性。因此，可以考虑采用增量式的层次聚类算法来解决分布式数据流的全局模式演化问题。为此，首先界定簇间的相似度[14]和相异度[15]。

定义 7(簇间相似度) 给定两个簇 C_a 和 C_b，对应的簇中心表示为 a 和 b，假如使用距离度量 d(如欧式距离)，那么簇 C_a 和簇 C_b 之间的相似度可以定义为 $sim(c_a,c_b)=1/d(a,b)$。

相应地，也可以把 $d(a,b)$ 作为簇 C_a 和 C_b 之间的相异度的度量。

算法 3 全局凝聚聚类算法 GAC(Global Agglomerative Clustering)

输入：局部节点个数 m；

t 时刻的所有局部簇的数据概要 SummaryLocal[1：m]；

全局最优簇的个数 k；

t−1 时刻的中心节点的全局簇的数据概要 SummaryCentral[1：k]；

全局簇集的凝聚度之和 Cohesion。

输出：对应 t 时刻，更新后的 SummaryCentral[1：k]和 Cohesion。

方法：

(1) 依据 t 时刻的 SummaryLocal[1：m]，使用算法 RSP 还原样本数据集，记 S；

(2) 对 S 聚类得到 k 个簇，记 C；

(3) 针对 C 和 t−1 时刻的 SummaryCentral[1：k]对应的簇，计算这 2k 个簇的相异度矩阵 M。

(4) 依据 M 计算凝聚度之和，记 Cohesion_{pre}；

(5) 从 M 中找出最小相异度的两个簇 c_a，c_b；

(6) 尝试合并 c_a 和 c_b 后再计算凝聚度之和，记 Cohesion_{new}；

(7) 假如 $\text{Cohesion}_{new}<\text{Cohesion}_{pre}$，则合并有效，更新 M；

(8) 否则取消合并，恢复 c_a 和 c_b，将 c_a 和 c_b 的相异度置为极大；

(9) 假如中心簇集的个数大于 k 或者 M 不全是极大值，转到(5)；

(10) 计算 SummaryCentral[1：k]和 Cohesion，结束。

4 实验结果及分析

实验的数据通过 Weka 平台定制，Weka 产生的数据依附随机函数生成的时间序列来模拟数据流的流动[16]。本文实验使用 $m=3$ 条局部数据流。采用时间数据窗口的方法来分块处理数据流[17]，大小为 $\delta=3000$。

首先，从聚类质量对 DGCM 进行整体评价。给定 k 个簇的集合 C，它的聚类质量可以表示为 $\Gamma(C)=B(C)/W(C)$，其中：$B(C)$ 和 $W(C)$ 代表簇间相异度 $B(C)$ 和簇内相异度 $W(C)$，分别被定义为下面的式(3)和式(4)：

$$B(C)=\sum_{1\leqslant i<j\leqslant k}d(C_i,C_j) \tag{3}$$

$$W(C)=\sum_{i=1}^{k}\sum_{x\in C_i}d(x,C_i) \tag{4}$$

很显然，$\Gamma(C)$ 越大，簇集 C 的质量越高。为了评价 DGCM，我们从它的聚类质量评价值 Γ、簇中心点位置以及数据传输代价等方面进行了实验和分析。

实验 1(DGCM 的聚类质量评估) 对比算法是改造后的 BirchCluster：在每个监测时刻首先将所有局部数据流的历史数据进行集成，然后使用 BirchCluster[10] 进行聚类。我们连续监控了 27 个窗口(约 8 万条记录)，图 1 给出了 DGCM 和 BirchCluster 聚类质量的对比。实验结果说明，DGCM 和 BirchCluster 的聚类质量相差无几，表明分布式的 DGCM 模型的聚类质量接近于先数据集成再挖掘的集中式 BirchCluster 方法对应的精度。

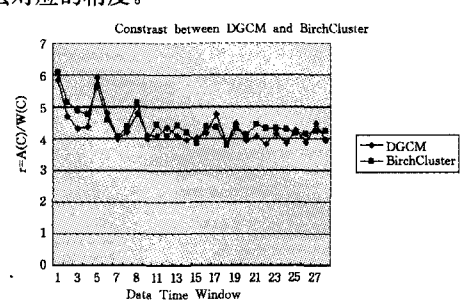

图 1 DGCM 与 BirchCluster 的聚类质量指标对比图

实验2(DGCM的簇中心跟踪)　仍然连续监控27个窗口(约8万条记录),BirchCluster算法[10]采用一次数据集成后进行聚类,而DGCM则采用窗口进行增量式聚类,选取DGCM的最后一次聚类结果和BirchCluster结果进行比较。图2给出了被观察的2维分布式数据流对应的比较结果。

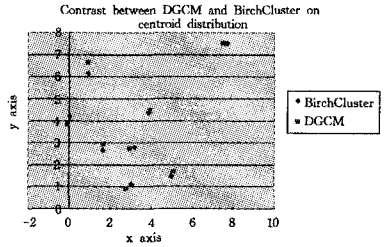

图2　DGCM与BirchCluster的簇中心分布(k=8)

图2显示,DGCM的最后一次聚类结果和数据集成的BirchCluster生成的簇中心非常接近,表明DGCM算法没有产生严重的簇中心偏差。

实验3(DGCM对高维数据的适应性)　数据维度采用2、7、12、17、22这5个值,不同维度下都是监控27个窗口(约8万条记录)数据。实验仍然是先将所有数据集成一起来运行BirchCluster算法[10],然后将其与本文提出的DGCM分布式挖掘模型进行对比实验。表2给出了两种方法在不同维度时发现的簇中心点的差异,其中两个算法的全局簇数目都是k=8。为了度量两个算法的簇中心点差异,采用最近簇中心点的最大距离差的计算方法,即从两种方法产生的簇中心点集合中,按照距离最近原则进行对比点的匹配,再把这k个对比点的距离最大值作为两种方法的簇中心点差异。

表2　BirchCluster和DGCM簇中心点差异对比

维度值	簇中心点差异
2	0.0024
7	0.0049
12	0.0352
17	0.1040
22	0.1537

表2说明随着数据维度增加,DGCM和BirchCluster聚类效果的差异会有所增加,但是差距并不大。当然,实验中BirchCluster是采用数据先集成再挖掘方法完成的,它的数据传输量是不可忽略的。

下面分析DGCM的数据传输量。

仅就实验1而言,改造后的BirchCluster需要在一个监测时间点,直接将所有节点的所有源数据传送到中心节点,进而获取全局模式。上传的总数据量为$Traffic(samples)=3*3000*T(tuple)$,$T(tuple)$代表一个数据元组的大小;而DGCM则只需在一个监测时间点将所有节点的数据概要传送到中心节点,上传的总数据量为$Traffic(synopsis)=3*k*T(synopsis)$,$T(synopsis)$代表数据概要的大小,$k$表示每个局部节点维持的簇的个数。本文的概要数据结构仅仅包含了6个基本属性,因此,$T(synopsis)\leqslant 6*T(tuple)$,假如$k$=8,那么两者通信量的比值"$Traffic(samples)$:$Traffic(synopsis)$"不小于"3000:48",即DGCM的传输代价要比直接源数据传输代价的1/50还要少。

结束语　本文设计了一种分布式数据流的聚类模型及与之配套的一系列算法。在局部节点上,改进最远点聚类算法、以增量式方式挖掘数据概要;只传输节点挖掘的数据概要以保证数据传输量足够小;在中心节点,采用层次聚类算法启发式挖掘来保证全局聚类质量。实验表明,该模型及其对应的算法在确保较小的数据通信代价的前提下,可以获得很高的挖掘效率和较高的挖掘质量。

参考文献

[1] Babcock B, Babu S, Datar M. Models and issues in data stream systems[C]// Proceedings of the 21st ACM Symposium on Principles of Database Systems. Madison, WI, USA: ACM, 2002: 1-16

[2] Khalilian M, Mustapha N. Data stream clustering: challenges and issues[C]// Proceedings of 2010 International MultiConference of Engineering and Computer Scientists. Hong Kong, China: Newswood Limited International Association of Engineers, 2010: 566-569

[3] Rajasegarar S, Leckie C, Palaniswami M. Distributed anomaly detection in wireless sensor networks[C]// Proceedings of the 10th IEEE Singapore International Conference on Communications Systems. Singapore, IEEE, 2006: 1-5

[4] Zhang Q, Liu J, Wang W. Approximate clustering on distributed data streams[C]// Proceedings of IEEE 24th International Conference on Data Engineering. Cancun, Mexico: IEEE, 2008: 1131-1139

[5] Graham C, Muthukrishnan S, Zhuang W. Conquering the divide: continuous clustering of distributed data streams[C]// Proceedings of the 23rd International Conference on Data Engineering. Istanbul, Turkey: IEEE, 2007: 1036-1045

[6] Hajiee M. A new distributed clustering algorithm based on K-means algorithm[C]// Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering. Piscataway. NJ, USA: IEEE, 2010: 2408-2411

[7] Januzai E, Kriegel H P, Pfeifle M. DBDC: density based distributed clustering[C]// Proceedings of Advances in Database Technology-EDBT 2004 9th International Conference on Extending Database Technology. Berlin, Germany: IEEE, 2004: 88-105

[8] Johnson E, Kargupta H. Collective, Hierarchical clustering from distributed, heterogeneous data[C]// Proceedings of 2000 Large-Scale Parallel Data Mining. London, UK: Springer-Verlag, 2000: 221-244

[9] Domingos P, Hulten G. Mining high-speed data streams [C]// Proceedings of KDD-2000 Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston, MA, USA: IEEE, 2000: 71-80

[10] Zhang T, Raghu R, Livny M. BIRCH: an efficient data clustering method for very large databases[J]. Sigmod Record, 1996, 25(2): 103-114

[11] Rodriques P P, Gama J, Lopes L. Clustering distributed sensor data streams[C]// Proceedings of Machine Learning and Knowledge Discovery in Databases. Antwerp, Belgium: Springer-Verlag, 2008: 282-297

[12] 郑铎,吴世伟. 正态分布函数计算的建议及其反函数的非迭代算法[J]. 河海大学学报:自然科学版,1993(02):61-64

[13] 朱晓玲,姜浩. 任意概率分布的伪随机数研究和实现[J]. 计算机技术与发展,2007,17(12):116-118

(下转第202页)

(2)对词条正文分词后的词项建立语言网络，再利用式(2)计算各个词项的综合特征值，按综合特征值从大到小排序后选择 TOP 比例的词项作为关键词建立向量空间，再利用式(1)计算词条正文之间的相似度，得到 Sim_2；

(3)对开放分类，利用式(3)计算词条之间的相似度，得到 Sim_3；

(4)对相关词条，直接建立向量空间后利用式(1)计算相似度，得到 Sim_4。

在得到词条各部分的相似度后，本文经过多次实验比较，根据关键词抽取规则[15]以及词条中百科名片和词条正文对词条意义贡献比较大、开放分类和相关词条对词条意义贡献较小的情况，选取的参数为 TOP＝30%，$\theta_1=0.4$，$\theta_2=0.4$，$\theta_3=0.1$，$\theta_4=0.1$，利用式(4)计算词条之间的相似度。本文使用文献[7]基于《知网》的方法和文献[8]基于《同义词词林》的方法作对比，得到的实验结果数据集如表 3 所列。

表 3　实验结果数据

中文词对	标准值	基于知网	基于同义词词林	本文算法
旅程-航程	0.929	0.04	0	0.63
货币-现金	0.908	1	0.43	0.57
计算机-软件	0.85	0.44	0.22	0.78
网络-硬件	0.831	0.29	0.22	0.54
自然-环境	0.831	0.05	0	0.62
心理学-弗洛伊德	0.821	0	0	0.61
新闻-报告	0.816	0.62	0.22	0.65
战争-部队	0.813	0.15	0.61	0.75
银行-货币	0.812	0.11	0.21	0.72
股票-市场	0.808	0.11	0.21	0.45
世纪-国家	0.316	0.11	0	0.21
志愿者-座右铭	0.256	0.1	0	0.15
原因-高血压	0.231	0.29	0.21	0.13
能源-秘书	0.181	0.10	0	0.08
股票-手机	0.162	0.26	0	0.11

从以上实验结果可以看到：(1)基于《知网》的相似度计算方法存在一些未登录词，对于一些新词以及一些不常用的词无法计算相似度。基于《同义词词林》的方法也同样存在一些未登录词，同时受到同义词词林层次关系的限制，得到的结果值集中于 5 个数值，不能很好地反映词间的语义关系。本文算法在这一点得到了很好的改进，能够计算出任意两个词语间的相似度值。(2)个别存在比较明显的概念关系的词语，其它两种方法的结果更优，但在整体效果与标准数据集相比方面，本文算法的结果显得更加合理有效。

结束语　本文提出了一种新的基于百度百科的词语相似度计算方法。与传统的基于语义词典和大规模语料库的方法不同，本文通过计算表征百科词条各个部分内容的相似度加权得到词条相似度。具体讨论了百科名片、词条正文、开放分类和相关词条部分的相似度计算方法，对其再加权就得到整体的相似度结果。从实验结果看，这个新方法产生的结果优于已有的方法。

本文后续的研究将在现有探讨词条相似度的基础上，进一步深入分析词条信息所蕴含的语义相似性特征，考虑百科名片、词条正文等语义结构信息，更好地提高词语相似度效果。

参 考 文 献

[1] 章志凌，虞立群，陈奕秋，等. 基于 Corpus 库的词语相似度计算方法[J]. 计算机应用，2006，26(3)：638-640，644

[2] Salton G，Lesk M E. Computer evaluation of indexing and text processing[J]. Journal of the ACM，1968，15(1)：8-36

[3] Rada R. Development and application of a metric on semantic nets[J]. IEEE Transactions on System. Man and Cybernetics，1989，19(1)：17-30

[4] Lee J H. Information retrieval based on conceptual distance in ISA hierarchies [J]. Journal of Documentation，1993，49(2)：188-207

[5] Agirre E，Rigau G. A Proposal for word sense disambiguation using conceptual distance [C]// International Conference/Recent Advances in Natural Language Recessing RANLP. 95. Tzigov Chark，Bulgaria，1995：91-98

[6] Sussna M. Word sense disambiguation for free-text indexing using a massive semantic network[C]// Proceedings of the 2nd International Conference on Information and Knowledge Management (CIKM'93). Washington，DC，US，1993：67-74

[7] 刘群，李素建. 基于《知网》的词汇语义相似度计算[C]// 台北第三届汉语词汇语义学研讨会

[8] 王斌. 汉英双语语料库自动对齐研究[D]. 北京：中国科学院计算技术研究所，1999

[9] Li Su-jian，et al. Semantic computation in Chinese question-answering system [J]. Journal of Computer Science and Technology，2002，17(6)：933-939

[10] Brown P. Word sense disambiguation using tactical methods[C]// Proceedings of 29th Meeting of the Association For Computational Linguistics (ACL291). 1991：210-207

[11] 胡俊峰，俞士汶. 唐宋诗词汇间语义相似度计算[J]. 中文信息学报，2002(4)：40-45

[12] Ferreri Cancho R，Sole R V. The small world of human language [J]. Biological Sciences，2001，268(1482)：2261-2265

[13] Seco N，Veale T，Hayes J. An Intrinsic Information Content Metric for Semantic Similarity in WordNet[C]// Proc of ECAI. 2004

[14] 黄承慧，印鉴，候昉，等. 一种结合词项语义信息和 TF-IDF 方法的文本相似度量方法[J]. 计算机学报，2011(5)：856-864

[15] 郑家恒，卢娇丽，等. 关键词抽取方法的研究[J]. 计算机工程，2005(9)：194-196

(上接第 191 页)

[14] O'Callaghan L，Mishra N，Meyerson A. Streaming-data algorithms for high-quality clustering[C]// Proceedings of 18th International Conference on Data Engineering. Los Alamitos，CA，USA：IEEE，2002：685-94

[15] Gorawski M，Pluciennik-Psota E. Distributed data mining methodology for clustering and classification model[C]// Proceedings of 10th International Conference on Artificial Intelligence and Soft Computing. Berlin，Germany：The Institution of Engineering and Technology，2010：323-30

[16] 孙岳，毛国君，刘旭. 基于多分类器的数据流中的概念漂移挖掘[J]. 自动化学报，2008，34(1)：93-97

[17] 吴枫，仲妍，吴泉源. 基于时间衰减模型的数据流频繁模式挖掘[J]. 自动化学报，2010，36(5)：674-684