基于相异度矩阵的混合属性数据流聚类算法

万仁霞,王立新,刘振文
WAN Ren-xia,WANG Li-xin,LIU Zhen-wen

东华大学 信息科学与技术学院,上海 201620
College of Information Science and Technology,Donghua University,Shanghai 201620,China
E-mail:wrx1022@mail.dhu.edu.cn

WAN Ren-xia,WANG Li-xin,LIU Zhen-wen.Novel algorithm for clustering heterogeneous data stream based on dissimilarity matrix.Computer Engineering and Applications,2008,44(25):149-151.

Abstract: Data stream clustering is an important issue in data stream mining.In this paper,a novel algorithm is presented for clustering data stream with heterogeneous attributes.It adopts dissimilarity instead of the common clustering distance,and an equivalent dissimilarity matrix is used in the clustering process.Then the empirical evidence of this algorithm's superiority over CluStream and HCluStream algorithms on the real data sets is given.

Key words: data stream;dissimilarity;cluster;heterogeneous attributes

摘 要:数据流的聚类是数据流挖掘的一个重要问题。提出一种针对混合属性的数据流聚类算法,它采用相异度来代替普通的聚类距离,并将等价相异度矩阵引入聚类过程。基于真实数据集的实验表明该算法比基地同类算法具有更好的聚类性能。

关键词:数据流;相异度;聚类;混合属性
DOI:10.3778/j.issn.1002-8331(2008)25-0149-03 **文章编号:**1002-8331(2008)25-0149-03 **文献标识码:**A **中图分类号:**TP311

1 引言

数据流就是大量连续到达的、潜在无限的数据的有序序列,这些数据是按时间顺序的、快速变化的、海量的和潜在无限的。由于数据流的数据量太大,因此不可能存储整个数据流或者对其扫描多次。数据流的计算只能使用有限的内存和有限的处理时间单遍扫描数据,并且,数据流可能是高度动态的,并随时间而演变。

近几年来,如何从数据流上获取知识的数据流挖掘已引起了业界的广泛关注。数据流聚类作为知识发现的一个重要内容得到了深入的研究。数据流聚类的一个经典算法是 Guha 等提出的 STREAM 算法[1-2],该算法源于中位数聚类,采用批处理方式,利用有限的空间和时间对数据流进行分层聚类,是单遍扫描的一种有效算法。然而,该算法既不考虑数据的演变也不考虑时间粒度的变化。聚类可能受控于旧的、过期的数据流。C.Aggarwal,J.Han 等人在文献[3]中提出了 CluStream 算法,该算法由在线和离线两部分构成,由于对数据的处理和更新是增量式的,因此,CluStream 算法不仅能给出整个数据流聚类的结果,还可以给出任意时间范围内的聚类结果,以及进行数据流的进化分析。CluStream 算法的框架得到了数据挖掘界的广泛认可,几年来许多研究者围绕这一框架开展了大量的工作。清华大学的杨春宇等基于 CluStream 算法的框架提出了对具有混合属性的数据流的聚类算法 HCluStream[4],该算法有效地解决了 CluStream 算法只能处理连续型属性的不足。但由于该算法在处理分类属性时将每个属性的取值都进行匹配,当分类属性很多或分类属性的阈值很大时将导致大量的时间消耗。本文将在此方面做进一步的讨论。

2 dCluStream 算法

本文将沿袭 CluStream 算法的框架思想,并引入相异度矩阵(dissimilarity matrix)[5]来分析处理聚类过程。将该算法命名为 dCluStream。

首先在文献[4]基础上定义本文的若干符号。设待处理数据流是一个数据序列 $\overline{X}_1\cdots\overline{X}_i\cdots$,其到达时间标签分别为 $T_1\cdots T_i\cdots$。每个样本具有 c 维连续属性与 b 维分类属性,表示为 $\overline{X}_i=\overline{C}_i:\overline{B}_i=[x_i^1,\cdots,x_i^c,y_i^1,\cdots,y_i^b]$,其中 $\overline{C}_i$ 是 C 维连续属性 $x_i^1,\cdots,x_i^c$ 构成的向量,$\overline{B}_i$ 是 b 维分类属性 $y_i^1,\cdots,y_i^b$ 构成的向量。分类属性 $y^p(1\leqslant p\leqslant b)$的全部可能取值 F^p,y^p 的第 $k(1\leqslant k\leqslant F^p)$种可能值记为 v_k^p。

定义 1 将在样本集$\{\overline{X}_1\cdots\overline{X}_n\}$上的微聚类表示为一个包含 $\sum_{p=1}^{b}F^p+2c+4$ 个元素的元组:$\overline{CFT}=(H,\overline{CF2}^x,\overline{CF1}^x,CF2^t,CF1^t,t_n,n)$,其中 H 是分类属性的频度直方图,包含 $\sum_{p=1}^{b}F^p$ 个元素,其第 p 行的第 k 个元素对应于第 p 个分类属性的第 k 个取值得频

作者简介:万仁霞(1975-),男,博士生,主要研究方向:数据挖掘和知识发现,智能控制;王立新(1966-),男,博士生,主要研究方向:软件工程,智能控制;刘振文(1974-),男,博士生,主要研究方向:服务组合。

收稿日期:2007-11-01 **修回日期:**2008-01-21

度,用公式表示为:$H_{(p,k)}\sum_{m=1}^{n}h_{p,k}^{m}$,其中 $h_{p,k}^{m}$ 表示第 m 个样本的第 p 个分类属性是否取值为 v_k^p:

$$h_{p,k}^{m}=\begin{cases}0, y_m^p\neq v_k^p\\ 1, y_m^p=v_k^p\end{cases}$$

$\overline{CF1}^x$ 与 $\overline{CF2}^x$ 分别是连续属性的一阶矩和二阶矩,它们的第 k 个项分别为 $\overline{CF1}^x{}_{(k)}=\sum_{m=1}^{n}x_m^k$,$\overline{CF2}^x{}_{(k)}=\sum_{m=1}^{n}(x_m^k)^2$;而 $CF1^t=\sum_{m=1}^{n}T_m$,$CF2^t=\sum_{m=1}^{n}(T_m)^2$ 分别表示样本时间标签的一阶矩和二阶矩;t_n 是微聚类的最后更新时间;n 为样本集中的样本数。

根据上面的定义,再分别定义样本与样本、样本与微簇、微簇与微簇间的相异度,如下:

样本 $\overline{X}_i$ 与样本 $\overline{X}_j$ 之间的相异度:

$$D(\overline{X}_i,\overline{X}_j)=\frac{\sum_{k,p=1}^{f}\delta_{ij}^{(k,p)}\lambda_{k,p}D_{ij}^{(k,p)}}{\sum_{k,p=1}^{f}\delta_{ij}^{(k,p)}}+\frac{\sum_{p=1}^{e}\delta_{ij}^{(p)}\lambda_pD_{ij}^{(p)}}{\sum_{p=1}^{e}\delta_{ij}^{(p)}}\qquad(1)$$

其中,对于样本 $\overline{X}_l(l=i$ 或 $j)$ 的第 p 个分类属性的第 k 个取值用二元体 $[y_l^{(k,p)},n_l^{(k,p)}]$ 表示。如果 $y_l^{(k,p)}$ 缺失或 $y_l^{(k,p)}=0$,且第 p 个分类属性是非对称二元的,则 $n_l^{(k,p)}=0$,否则 $n_l^{(k,p)}=1$。如果 $n_i^{(k,p)}=0$ 或 $n_j^{(k,p)}=0$,则 $\delta_{ij}^{(k,p)}=0$,否则 $\delta_{ij}^{(k,p)}=1$;而 $\delta_{ij}^{(p)}$ 恒取值 1,因而 $\sum_{p=1}^{e}\delta_{ij}^{(p)}=e$;$\lambda_{k,p}$ 为样本第 p 个分类属性的第 k 个取值的聚类权重,λ_p 为样本第 p 个连续属性的聚类权重,且 $f=\sum_{p=1}^{b}F^p$,$\sum_{k,p=1}^{f}\lambda_{k,p}+\sum_{p=1}^{e}\lambda_p=1$;对于连续属性:$D_{ij}^{(p)}=\frac{|x_{ip}-x_{jp}|}{\max_{R_p}-\min_{R_p}}$,其中 R_p 为第 p 个连续属性的所有取值;对于分类属性:$D_{ij}^{(k,p)}=\begin{cases}1, y_i^{(k,p)}\neq y_j^{(k,p)}\\ 0, y_i^{(k,p)}=y_j^{(k,p)}\end{cases}$。

为了度量样本 $\overline{X}_i$ 与微簇 M_j 之间的相异度,先将微簇 M_j 用一虚拟样本 $\overline{X}_j'$ 替代,$\overline{X}_j'=(\overline{C}_j',\overline{B}_j')=(\frac{M_j\cdot\overline{CF1}^x}{M_j\cdot n},M_j\cdot H)$,其中 $\frac{M_j\cdot\overline{CF1}^x}{M_j\cdot n}$ 是 M_j 的连续属性的中心,对应的虚拟样本 $\overline{X}_j'$ 的连续属性的值,$M_j\cdot H$ 是 M_j 的分类属性的频度直方图,其对应的虚拟样本 $\overline{X}_j'$ 的第 p 个分类属性第 k 个值为 $[y_{j_0}^{(k,p)},n_{j_0}^{(k,p)}]$ 当且仅当其出现的频数 $n_{j_0}^{(k,p)}=\underset{\|M_j\|}{\mathrm{Max}}\{n_j^{(k,p)}\}$,其中 $n_j^{(k,p)}$ 为 M_j 中所有样本的第 p 个分类属性第 k 个取值 $y_j^{(k,p)}$ 的计数,则样本 $\overline{X}_i$ 与微簇 M_j 之间的相异度可表示为:$D(\overline{X}_i,M_j)=D(\overline{X}_i,\overline{X}_j')$,再调用公式(1),即得到样本与微簇间的相异度公式.

至于计算微簇 M_i 与微簇 M_j 之间的相异度,其原理与样本与微簇之间相异度的计算类似,即首先分别给微簇 M_i 与微簇 M_j 寻找替代它们的虚拟样本 $\overline{X}_i'$ 与 $\overline{X}_j'$,然后计算 $\overline{X}_i'$ 与 $\overline{X}_j'$ 之间的相异度,即得微簇 M_i 与微簇 M_j 之间的相异度。

所以,整个微聚过程的核心还是样本间的聚类。

虽然数据流的数据量本身可能是巨大的,但是,由于在给定时间戳上总能根据需要将到达的数据流分成几个有限数据的块(chunk)[1],而每个块上的样本也总是有限的。下面讨论有限数据块上的有效微聚过程。

定义 2 将所有样本间的相异度表示成一个 $n\times n$ 的矩阵:

$$\begin{bmatrix}0 & & \cdots & \\ d(2,1) & 0 & & \cdots \\ d(3,1) & d(3,2) & 0 & \cdots \\ \vdots & \vdots & \vdots & \\ d(n,1) & d(n,2) & d(n,3) & \cdots\ 0\end{bmatrix}$$

称为相异度矩阵。其中,$d(i,j)$ 表示样本 $\overline{X}_i$ 与样本 $\overline{X}_j$ 之间的相异度,即 $d(i,j)=D(\overline{X}_i,\overline{X}_j)$。

由于 $d(i,j)=d(j,i)$,并且 $d(i,j)=0$,所以相异度矩阵是对称的、自反的。

定义 3 设 $[d_1(i,j)]_{n\times n}$ 和 $[d_2(i,j)]_{n\times n}$ 是两个相异矩阵,定义相异矩阵的合成运算、并运算和包含关系:

合成运算:$[d_1(i,j)]_{n\times n}\cdot[d_2(i,j)]_{n\times n}=[d_3(i,j)]_{n\times n}$,其中 $d_3(i,j)=\bigwedge_{k=1}^{n}(d_1(i,k)\vee d_2(k,j))$。

并运算:$[d_1(i,j)]_{n\times n}\cup[d_2(i,j)]_{n\times n}=[d_4(i,j)]_{n\times n}$,其中 $d_4(i,j)=d_1(i,j)\wedge d_2(i,j)$。

包含关系:如果任意 i,j,都有 $d_1(i,j)\leqslant d_2(i,j)$,则称 $[d_1(i,j)]_{n\times n}$ 包含 $[d_2(i,j)]_{n\times n}$。

定义 4 设 $[d(i,j)]_{n\times n}$ 是相异矩阵,如果 i,j,都有 $d(i,j)\leqslant\bigwedge_{k=1}^{n}(d(i,k)\vee d(k,j))$ 则称相异度矩阵 $[d(i,j)]_{n\times n}$ 是传递的。

根据文献[6]可知,若 $[d(i,j)]_{n\times n}$ 是相异度矩阵,则 $[d(i,j)]_{n\times n}^{n-1}$ 一定是等价的。基于此,将新算法的微聚过程描述如下:

第一步,根据之前样本间相异度的定义计算相异度矩阵 $[d(i,j)]_{n\times n}$ 中元素 $d(i,j)$ 的值;

第二步,利用逐次平方法,计算 $[d(i,j)]_{n\times n}\to[d(i,j)]_{n\times n}^{2}\to\cdots\to[d(i,j)]_{n\times n}^{2^k}$ 直到首次出现 $2^k\geqslant n-1$,此时 $[d(i,j)]_{n\times n}^{n-1}=[d(i,j)]_{n\times n}^{2^k}$;

第三步,给定相异度水平 η,如果 $d^{(n-1)}(i,j)\leqslant\eta$,则样本 $\overline{X}_i$ 与样本 $\overline{X}_j$ 归为同一类。

对于不同阶段的微聚,相异度水平 η 应该是不一样的,将样本与样本、样本与微簇、微簇与微簇间的相异度水平区分为 η_1、η_2、η_3。理论上相异度水平 η 取值越小,保留的中间聚类信息就越丰富,就越有利于最终的聚类效果。

在宏聚阶段,首先将文献[3]中提出的宏聚的两个重要性质推广到当前的算法中:

性质 1 设 C_1 和 C_2 是两个点集,则 $\overline{CFT(C_1\cup C_2)}=\overline{CFT(C_1)}+\overline{CFT(C_2)}$。

性质 2 设 C_1 和 C_2 是两个点集且 $C_1\supseteq C_2$,则 $\overline{CFT(C_1-C_2)}=\overline{CFT(C_1)}-\overline{CFT(C_2)}$。

根据阶矩的特性,上述性质是显然的。

在这一阶段,首先根据用户指定的时间窗口 h 和相异度水平 η 找到当前快照 t_c 和离该窗口最近的快照 $t_s(t_s=t_c-h)$,再利用的微簇的可减性,将当前快照 t_c 中的各微簇减去 t_s 中的各微簇,这样就得了在时间 (t_c-h,t_c) 内各数据样本点的微簇集合,最后将各微簇采用之前所述的技术处理成相应的虚拟样本点,并将等价相异度矩阵作用在这些虚拟样本点上进行聚类,从而获

得用户想要的结果。

3 实验结果及结论

本文对所提出的 dCluStream 算法进行了性能测试。实验平台配置如下:CPU 为 Intel Pentium 1.7 GHz,内存为 512 MB,操作系统为 Windows XP Professional Edition,所用代码均用 C++ 编程实现。

本文的实验数据集采用 KDD-CUP'99 网络入侵侦查数据集,该数据集共有 494 020 个样本,每个样本含有 34 个数值属性、7 个分类属性。本文采用该数据集进行实验分析。

由于传统的基于距离平方和(SSQ)的质量评价方式只适合连续属性的数据,并不适用于新算法 dCluStream 的质量评价,因此将采用文献[7]中使用的聚类纯度以及聚类时间两个方面来比较 dCluStream 与 HCluStream、CluStream 性能质量。实验时以经验值 $\eta_1:\eta_2:\eta_3=1:2:4$ 的比例来确定对应的相异度水平,并给予每个属性相同的权重。

在做聚类纯度比较实验时由于本文采用的网络入侵侦查数据集,而网络入侵可大致概括为四种类型[1,3],因此实验时将聚类的宏簇数预定为 5 个(含未受攻击入侵)。

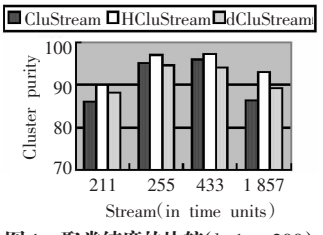

图 1 聚类纯度的比较(h=1,s=200)

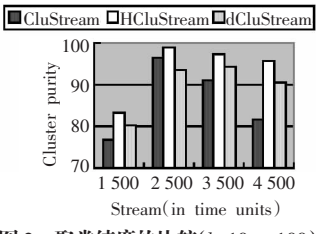

图 2 聚类纯度的比较(h=10,s=100)

在图 1 和图 2 中,dCluStream 总体上比 CluStream 具有更可靠的聚类纯度,而 HCluStream 的聚类纯度又比 dCluStream 稍好些,这是因为聚类时 CluStream 算法完全丢弃了分类属性,而 dCluStream 算法在进行相异度矩阵的合成或并运算时是有信息损失的,这就造成了 CluStream,dCluStream 都会比 HCluStream 有稍逊的聚类纯度。

为了研究算法的聚类时间,考察了不同簇数的条件下,dCluStream、HCluStream 与 CluStream 的聚类时间,如图 3。

在图 3 中,dCluStream 有与 CluStream 算法几乎相当的运行时间,而 HCluStream 相比之下是很费 CPU 时间的。原因在于 HCluStream 在处理分类属性时大量的匹配运算导致 CPU 时间的大量浪费,而 dCluStream 与 CluStream 都以各自的方式压缩了聚类信息,并以该压缩的信息进行了聚类。

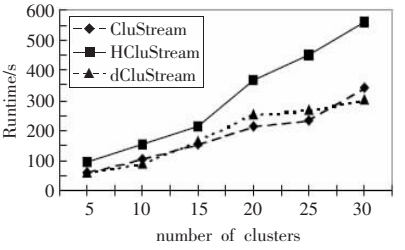

图 3 聚类时间的比较(h=1,s=2 000)

通过上述实验表明,dCluStream 算法以一定的信息损失为代价换来 HCluStream 算法在 CPU 时间上过多消耗的改善,同时又克服了 CluStream 算法只能处理连续属性的不足。由于流数据处理时间的有限性,因此,dCluStream 算法是混合属性流聚类算法中对于 HCluStream 算法时间上的有效折衷。

4 未来工作展望

由于 dCluStream 算法在聚类前须指定相异度水平 η,不同的相异度水平会得到不同的聚类效果。然而在聚类前指定相异度水平 η 要比预先指定聚类簇数的传统聚类算法要难以操作的多。如何指定一个合适的相异度水平 η 及其聚类临界值的确定都需要进一步的研究。另外,在构造等价相异度矩阵时造成的信息损失在多大程度上是可接受的这一问题也有待于深入讨论。这些都是下一步需要开展的工作。

参考文献:

[1] Guha S,Mishra N,Montwani R,et al.Clustering data streams[C]// Proc of IEEE Symposium on Foundations of Computer Science (FOCS'00),2000:71-80.

[2] Guha S,Meyerson A,Mishra N,et al.Clustering data streams:Theory and practice[J].IEEE Transactions on Knowledge and Data Engineering,2003,15(3):515-528.

[3] Aggarwal C,Han J,Wang J,et al.A framework for clustering evolving data stream[C]//Proc of Int Conf on Very Large Data Bases (VLDB'03),2003:81-92.

[4] 杨春宇,周杰.一种混合属性数据流聚类算法[J].计算机学报,2007,30(8):1364-1371.

[5] Han J,Kamber M.Data Mining:concepts and techniques[M].2nd ed.[S.l.]:Morgan Kaufman,2006:386-397.

[6] 赵明清,蒋昌俊,陶树平.基于等价相异度矩阵的聚类[J].计算机科学,2004,31(7):183-184.

[7] Aggarwal C,Han J,Wang J,et al.A framework for projected clustering of high dimensional data stream[C]//Proc of Int Conf on Very Large Data Bases(VLDB'04),2004:852-863.

(上接 145 页)

管理学院刘丽萍博士、山西大学计算机与信息技术学院王振强同学、信息产业部第三十三研究所刘伟同志、太原理工大学生计算机与软件学院郭浩同学和柴忠同学表示感谢。

参考文献:

[1] 李向阳,苗壮.自由文本信息抽取技术[J].情报科学,2004,2(7):815.

[2] 李向阳,张亚非.基于语义标注的信息抽取[J].解放军理工大学学报,2004,5(4):39.

[3] 由丽萍.构建现代汉语框架语义知识库技术研究[D].上海:上海师范大学,2006.

[4] Fillmore C J.框架语义学[M].詹卫东,译.北京:商务印书馆,2003.

[5] 李保利,陈玉忠,俞士汶.信息抽取研究综述[J].计算机工程与应用,2003,39(5):61.

[6] 李跃进,林鸿飞.基于 Internet 军事演习信息抽取系统[J].计算机工程与应用,2006,42(14):214-218.

[7] 姜吉发.自由文本的信息抽取模式获取的研究[D].北京:中国科学院研究生院,2004.

[8] Ciravegna F.Adaptive information extraction from text by rule induction and generalisation[D].Department of Computer Science, University of Sheffield.