BIRCH 混合属性数据聚类方法

李 贤,罗 可
LI Xian,LUO Ke

长沙理工大学 计算机与通信学院,长沙 410004
Department of Computer and Communication,Changsha University of Science & Technology,Changsha 410004,China
E-mail:lx20010@gmail.com

LI Xian,LUO Ke.Heterogeneous data clustering algorithm of BIRCH.Computer Engineering and Applications,2009,45(30):123-125.

Abstract: Data clustering is an important issue in data mining.Many real-world data have both continuous attributes and categorical attributes,which are usually called heterogeneous attributes.However,most of the existing mining algorithms can manipulate only continuous attributes or categorical attributes.Simply omitting categorical or continuous attributes may lose important information about the data and decrease the mining quality.Some other algorithms which can manipulate continuous attributes and categorical attributes have low efficiency,because of a lot of attributes.This paper proposes a novel approach for clustering data with heterogeneous features based on BIRCH.Experimental results on public data sets show that the proposed algorithm is robust.
Key words: data mining;clustering;BIRCH algorithm;heterogeneous attribute

摘 要:数据聚类是数据挖掘中的重要研究内容。现实世界中的数据往往同时具有连续属性和离散属性,但现有大多数算法局限于仅处理其中一种属性,而对另一种采取简单舍弃的办法丢失聚类信息和降低聚类质量。一些能处理混合属性的算法又往往处理的属性过多,导致计算量的大增。提出了一种基于 BIRCH 算法的混合属性数据的聚类算法;在 UCI 数据集上的实验表明,文中提出的算法具有较好的性能。
关键词:数据挖掘;聚类;BIRCH 算法;混合属性
DOI:10.3778/j.issn.1002-8331.2009.30.038 **文章编号:**1002-8331(2009)30-0123-03 **文献标识码:**A **中图分类号:**TP31

1 引言

近年来,随着计算机技术、通信技术以及网络技术的飞速发展,许多领域中出现了各种复合属性的数据。典型例子包括电信呼叫数据、股票交易数据、网站访问日志、互联网通信数据、搜索引擎数据、大型零售企业销售数据等等[1]。

数据管理与分析是数据挖掘领域研究的热点之一,其中研究如何从数据中获取知识的数据挖掘更是获得了广泛的关注[2]。在众多数据挖掘任务中,数据聚类作为知识发现的重要手段得到了深入研究。当前主要的聚类算法有:(1)基于划分的方法,如 K 平均算法[3]和 K 中心点算法[4];(2)基于层次的方法,如 CURE[5]和 BIRCH[6-7];(3)基于密度的方法,如 DBSCAN 和 OPTICS;(4)基于网格的方法,如 STING、CLIQUE 和 WaveCluster;(5)基于模型的方法,如 COBWEB 等。目前已有的数据聚类算法大部分局限于处理只具有连续属性的数据,另外有少量的算法局限于处理只具有离散属性的数据。其中所谓的连续属性是指属性的取值为连续数值,如长度、重量;所谓的离散属性是指属性的取值为有限的状态,如颜色、职业。现实世界中的许多数据同时具有连续属性和离散属性,如网络数据包等。处理一类属性的算法在混合属性条件下必然损失数据信息,影响数据挖掘的质量,而一些能处理混合属性的算法 如 K-protoypes 在对大量数据处理时效率不理想。针对以上提出了一种基于分层聚类 BIRCH 算法的适用于处理混合属性数据的聚类算法——H-BIRCH(Heterogeneous BIRCH)。

2 研究背景

2.1 混合属性聚类研究现状

Z.Huang 提出的 k-mode 算法和 k-protoypes 算法推广 k-means 方法,使之可以对类属性和混合型属性的对象集进行聚类。陈宁、陈安和周龙骧进一步提出了模糊 k-protoypes 算法,并通过引进模糊算法以提高聚类的准确性。但 k-protoypes 方法及其改进型对大量数据处理时效率不理想。

2.2 BIRCH 算法

BIRCH 是一个综合的层次聚类算法, 它用聚类特征和聚类特征树(CF)来概括聚类描述,描述如下。

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.10826099,No.10871031);湖南省科技计划项目基金(the Research Foundation of Science and Technology Plan Project in Hunan Province of China under Grant No.2008FJ3015);湖南省教育厅科研项目基金(the Research Foundation of Department of Education in Hunan Province of China under Grant No.07A001)。
作者简介:李贤(1982-),男,硕士研究生,主要从事数据挖掘、计算智能研究;罗可(1961-),男,教授,博士,主要研究方向:数据挖掘、计算机应用。
收稿日期:2009-04-27 **修回日期:**2009-06-17

定义 1 对于具有 N 个 d 维数据点的簇 $\{X_i\}(i=1,2,3,\cdots,N)$，它的聚类特征向量定义为 $\boldsymbol{CF}=(N,LS,SS)$ 其中 N 为簇中点的个数；LS 表示 N 个点的线性和 $\left(\sum_{i=0}^{N}O_i\right)$，反映了簇的重心，$SS$ 是数据点的平方和 $\left(\sum_{i=0}^{N}O_i^2\right)$，反映了类直径的大小。

此外，对于聚类特征有如下定理。证明请见参考文献[4]。

定理 1 假设 $\boldsymbol{CF}1=(N1,LS1,SS1)$ 与 $\boldsymbol{CF}2=(N2,LS2,SS2)$ 分别为两个类的聚类特征，合并后的新类特征为 $\boldsymbol{CF}1+\boldsymbol{CF}2=(N1+N2,LS1+LS2,SS1+SS2)$。该算法通过聚类特征可以方便地进行中心、半径、直径及类内、类间距离的运算。CF 树是一个具有两个参数分支因子 B 和阈值 T 的高度平衡树，它存储了层次聚类的聚类特征。分支因子定义了每个非叶节点孩子的最大数目，而阈值给出了存储在树的叶子节点中的子聚类的最大直径。CF 树可以动态地构造，因此不要求所有的数据读入内存，而可在外存上逐个读入数据项。一个数据项总是被插入到最近的叶子条目（子聚类）。如果插入后使得该叶子节点中的子聚类的直径大于阈值，则该叶子节点极可能有其他节点被分裂。新数据插入后，关于该数据的信息向树根传递。BIRCH 算法通过一次扫描就可以较好地进行聚类，故该算法的计算复杂度是 $O(n)$，n 是对象的数目。

3 H-BIRCH 算法介绍

3.1 距离定义

定义 2 首先定义若干符号。待处理数据为 $\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_i,\cdots$ 每一个样本具有 b 维连续属性与 c 维离散属性，表示为 $\boldsymbol{X}_i=\boldsymbol{C}_i:\boldsymbol{B}_i=(x_i^1,x_i^2,\cdots,x_i^c,y_i^1,y_i^2,\cdots,y_i^b)$，其中 $\boldsymbol{C}_i$ 是由 c 维连续属性 $x_i^1,x_i^2,\cdots,x_i^c$ 构成的向量，$\boldsymbol{B}_i$ 是由 b 维离散属性 $y_i^1,y_i^2,\cdots,y_i^b$ 构成的向量离散属性，$y_p(1\leqslant p\leqslant b)$ 的全部可能取值数记为 F^p，y_p 的第 $k(1\leqslant k\leqslant F^p)$ 种可能值记为 V_k^p。

在样本集 $\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_i,\cdots\}$ 上定义 H 是离散属性的频度直方图，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 个离散属性的第 k 个取值的频度，可以用公式记作 $\boldsymbol{H}_{(p,k)}=\sum_{m=1}^{n}h_{p,k}^m$，其中 $h_{p,k}^m$ 表示第 m 个样本的第 p 个离散属性是否取值为 V_k^p，如式(1)所示

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

$\boldsymbol{F}$ 与 $\boldsymbol{D}$ 分别是连续属性的一阶矩和二阶矩，均包含 c 个元素，且 $\boldsymbol{D}^k=\sum_{m=1}^{n}(x_m^k)^2,\boldsymbol{F}^k=\sum_{m=1}^{n}x_m^k\quad(1\leqslant k\leqslant c)$。

一个数据可分为许多个不相交的样本子集，每一个样本子集 $\boldsymbol{C}_j$ 上的微聚类记作 $M_j=CFT(C_j)$，微聚类的属性用“·”表示，如 $M_j\cdot n$ 表示微聚类 M_j 包含的样本数量。为描述微聚类过程，首先基于文献[8]给出混合属性下的样本与样本之间、样本与微聚类之间以及微聚类与微聚类之间的距离定义。其中每一维连续属性均采用了文献[9]中提出的方式归一化，使其方差为 1。样本与样本的距离定义为：

$$Dss(\boldsymbol{X}_i,\boldsymbol{X}_j)=Dss(\boldsymbol{C}_i,\boldsymbol{C}_j)+\beta Dss(\boldsymbol{B}_i,\boldsymbol{B}_j)\tag{2}$$

其中 $Dss(\boldsymbol{C}_i,\boldsymbol{C}_j)=\sqrt{\sum_{k=1}^{c}(x_i^k-x_j^k)^2}$ 是连续属性部分的距离；$Dss(\boldsymbol{B}_i,\boldsymbol{B}_j)=\sum_{k=1}^{b}\delta(y_i^k,y_j^k)$ 是离散属性部分的距离，其中

$$\delta(y_i^k,y_j^k)=\begin{cases}1, & y_i^k\neq y_j^k\\ 0, & y_i^k=y_j^k\end{cases}\tag{3}$$

β 是离散属性部分的权重，样本与微聚类之间的距离定义为：

$$Dsm(\boldsymbol{X}_i,\boldsymbol{M}_j)=Dss(\boldsymbol{C}_i,\frac{\boldsymbol{M}_j\cdot\boldsymbol{F}}{\boldsymbol{M}_j\cdot\boldsymbol{n}})+\beta Dsm(\boldsymbol{B}_i,\boldsymbol{M}_j\cdot\boldsymbol{H})\tag{4}$$

其中 $(\boldsymbol{M}_j\cdot\boldsymbol{F})/(\boldsymbol{M}_j\cdot\boldsymbol{n})$ 是 $\boldsymbol{M}_j$ 连续属性的中心，$Dsm(\boldsymbol{B}_i,\boldsymbol{M}_j\cdot\boldsymbol{H})$ 是离散属性部分距离，简记为 $Dsm(\boldsymbol{B}_i,\boldsymbol{H})$

$$Dsm(\boldsymbol{B}_i,\boldsymbol{H})=\sum_{p=1}^{b}\sum_{k=1}^{F^p}\left[\frac{\delta(y_i^p,v_k^p)\boldsymbol{H}_{(p,k)}}{\sum_{m=1}^{F^p}\boldsymbol{H}_{(p,m)}}\right]\tag{5}$$

微聚类与微聚类之间的距离定义为

$$Dmm(\boldsymbol{M}_i,\boldsymbol{M}_j)=Dss(\frac{\boldsymbol{M}_i\cdot\boldsymbol{F}}{\boldsymbol{M}_i\cdot\boldsymbol{n}}\cdot\frac{\boldsymbol{M}_j\cdot\boldsymbol{F}}{\boldsymbol{M}_j\cdot\boldsymbol{n}})+\beta Dmm(\boldsymbol{M}_i\cdot\boldsymbol{H},\boldsymbol{M}_j\cdot\boldsymbol{H})\tag{6}$$

其中 $Dss((\boldsymbol{M}_i\cdot\boldsymbol{F})/(\boldsymbol{M}_i\cdot\boldsymbol{n}),(\boldsymbol{M}_j\cdot\boldsymbol{F})/(\boldsymbol{M}_j\cdot\boldsymbol{n}))$ 是连续属性部分的距离，$Dmm(\boldsymbol{M}_i\cdot\boldsymbol{H},\boldsymbol{M}_j\cdot\boldsymbol{H})$ 是离散属性部分的距离，用 $Dmm(\boldsymbol{H}^i,\boldsymbol{H}^j)$ 替代

$$Dmm(\boldsymbol{H}^i,\boldsymbol{H}^j)=b-\sum_{p=1}^{b}\frac{\boldsymbol{H}_p^i\cdot\boldsymbol{H}_p^{jT}}{\|\boldsymbol{H}_p^i\|_2\|\boldsymbol{H}_p^j\|_2}\tag{7}$$

其中，$\boldsymbol{H}_p$ 表示 H 的第 p 行，即称离散属性 p 的取值频度向量。

根据以上所述，聚类特征向量不再是 $\boldsymbol{CF}=(N,LS,SS)$ 需要修改聚类特征向量，修改 CF 结构为 H-CF 结构，H-CF 定义为

$$\boldsymbol{H\text{-}CF}=\{\boldsymbol{M}_J\cdot\boldsymbol{n},\boldsymbol{M}_J\cdot\boldsymbol{H}_{(p,q)},\boldsymbol{M}_J\cdot\boldsymbol{F}^k,\boldsymbol{M}_J\cdot\boldsymbol{D}^k\}$$

3.2 算法描述

步骤 1 将数据逐个输入。

步骤 2 寻找输入点与当前微聚类的最近距离 $Dss_{\min}$ 或者 $Dsm_{\min}$，并加入此微聚类 C_j。

步骤 3 如果 C_j 微聚类直径小于阈值 T 将 C_j 微聚类 H-CF 结构更新。

$$\begin{cases}\boldsymbol{M}_J\cdot\boldsymbol{H}_{(p,q)}=\boldsymbol{M}_J\cdot\boldsymbol{H}_{(p,q)}+h_{p,q}\\ \boldsymbol{M}_J\cdot\boldsymbol{n}=\boldsymbol{M}_J\cdot\boldsymbol{n}+1\\ \boldsymbol{M}_J\cdot\boldsymbol{F}^k=\boldsymbol{M}_J\cdot\boldsymbol{F}^k+x^k\\ \boldsymbol{M}_J\cdot\boldsymbol{D}^k=\boldsymbol{M}_J\cdot\boldsymbol{D}^k+(x^k)^2\end{cases}$$

如果大于则转到步骤 4。

步骤 4 此时 C_j 微聚类直径大于阈值 T，分裂微聚类 C_j，分裂的原则是寻找该叶节点中距离最远的两个条目并以这两个条目作为分裂后新的两个叶节点的起始条目，其他剩下的条目根据距离最小原则分配到这两个新的叶节点中，删除原叶节点并更新整个 H-CF 树。

步骤 5 最后根据 T 和 B 生成 R 个微聚类。进入宏聚类阶段。

步骤 6 宏聚类阶段开始，先将微聚类中数据点过少的微聚类当作异常点或噪声点去除。

步骤 7 根据每个微聚类之间的距离 Dmm，不断合并最相近的微聚类直到达到所需聚类数 K。

4　实验结果

全部实验均在 PC 上完成。所用计算机配置如下：CPU 为 InterPentiumⅣ3.0 GHz，内存为 1 GB DDRROM，操作系统为 Window XP，编程语言为 C++。

实验数据集与文献[8]中所用标准数据集一致，为 Forerst-CoverType 数据集。这个数据集为 UCI 数据挖掘数据集 ForerstCoverType 是美国森林服务信息系统提供的数据，共包含 581 012 条记录。每条记录是一块面积为 9.144 m×9.144 m 土地上的地理数据，包含 54 维属性，其中连续属性 10 维，离散属性 44 维，对应于 7 种森林覆盖类型。

采用了文献[8]中使用的聚类纯度做为对比 H-BIRCH 和 K-Protoypes 的度量，从质量和效率两方面来衡量算法。

H-BIRCH 的时间复杂度为 $O(n)$，而 K-Protoypes 的时间复杂度为 $O(M\cdot N^k)$，其中 M 为迭代次数，K 为聚类中心数目，实验显示 H-BIRCH 的效率要大大优于 K-Protoypes 的效率。从实验的结论来看也说明了这样的一个问题。

从图 1 中可以看出 K-Protoypes 为指数增长，H-BIRCH 为线性增长。

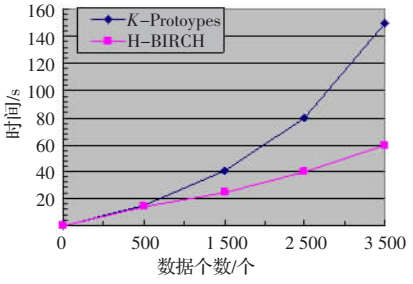

图 1　K-Protoypes 算法与 H-BIRCH 算法效率比较

从图 2 中可以看出 H-BIRCH 比 K-Protoypes 的聚类质量要略好，由于本数据集是一接近球形的数据集，所以 H-BIRCH 方法对此数据级有着不错的聚类质量，对球形数据集 H-BIRCH 算法可以在不大幅降低聚类质量的情况下，大幅提高混合属性数据聚类的速度，适合混合属性多，数据样本大，并呈球形分布的数据集。

5　结论

当对球形数据集时 H-BIRCH 有着不错的处理算法效率，

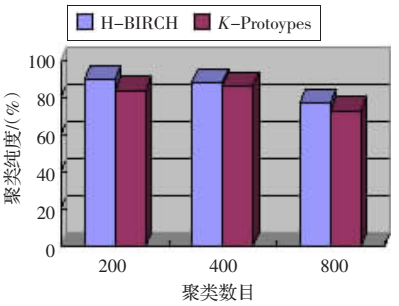

图 2　H-BIRCH 与 K-Protoypes 算法聚类纯度比较

这样可以将 H-BIRCH 应用到有着大量的混合属性的 Web 聚类，将 H-BIRCH 应用到 Web 聚类，并解决对非球形数据的聚类将是下一步工作。

参考文献：

[1] Muthukrishnan S.Data streams：Algorithms and apppication[M]. Hanover，MA，USA：Now Publishers Inc，2005.

[2] Gaber M M，Zaslavsky A B，Krishnaswamy S.Mining data Streams：A review[J].SIGMOD Record，2005，34(2)：18-26.

[3] Kaufan L，Rousseeuw P J.Finding groups in data：An introduction to cluster analysis[M].New York：John Wiley&Sons，1990.

[4] Han J W.Kambr M.Data mining concepts and techniques[M].Beijing：Higher Education Press，2001：145-176.

[5] Guha S，Rastogi R，Shim K.CURE：An efficient clustering alo-gorithm for large database[C]//Haas L M，Tiwary A.Proceedings of the ACM SIGMOD International Conference on Management of Data. Seattle：ACM Press，1998：73-84.

[6] 忻凌，倪志伟，黄玲.基于数据流的 BIRCH 改进聚类算法[J].计算机工程与应用，2007，43(5)：166-168.

[7] 蒋盛益，李霞.一种改进的 BIRCH 聚类算法[J].计算机应用，2009，29(1)：293-296.

[8] 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)：1364-1372.

[9] Aggarwal C C，Han Jia-wei，Wang Jian-yong，et al.A framework for projected clustering of high dimensional data streams[C]//Proceedings of the 30th Inernational Conference on Very Large Data Bases，Toronto，Canada，2004：852-863.

（上接 117 页）

参考文献：

[1] Sae-Young C，Forney G D，Richardson Jr T J，et al.On the design of low-density parity-check codes within 0.004 5 dB of the Shannon limit[J].IEEE Communications Letters，2001，5：58-60.

[2] Wang Z，Cui Z.A memory efficient partially parallel decoder architecture for quasi-cyclic LDPC codes[J].IEEE Transactions on Very Large Scale Integration(VLSI) Systems，2007，15(4)：483-488.

[3] Darabiha A，Carusone A C.Block-interlaced LDPC decoders with reduced interconnect complexity[J].IEEE Transactions on Circuits and Systems II：Express Briefs，2008，55(1)：74-78.

[4] Hao Z，Tong Z.Quasi-cyclic LDPC codes for the magnetic recording channel：Code design and VLSI implementation[J].IEEE Transactions on Magnetics，2007，43(3)：1118-1123.

[5] Kachinschang F R，Frey B J，Loeliger H A.Factor graphs and the sum-product algorithm[J].IEEE Transactions on Information Theory，2001，47(2)：498-519.

[6] Fossorier M，Valembois A.Reliability-based decoding of Reed-Solomon codes using their binary image[J].IEEE Communication Letters，2004，8(7)：452-454.

[7] Fossorier M P C.Iterative reliability-based decoding of low-density parity check codes[J].IEEE Journal on Selected Areas in Communications，2001，19(5)：908-917.

[8] Fossorier M，Lin S.Soft decision decoding of linear block codes based on ordered statistics[J].IEEE Transactions on Information Theory，1995，41(5)：1379-1396.

[9] Chase D.A class of algorithms for decoding block codes with channel measurements information[J].IEEE Transactions on Information Theory，1972，18(1)：170-182.

[10] Gallager R G.Low density parity-check codes[J].IEEE Transactions on Information Theory，1962，8(1)：21-28.