

# 一种高效的数据流挖掘增量模糊决策树分类算法

王 涛<sup>1)</sup> 李舟军<sup>2)</sup> 胡小华<sup>3)</sup> 颜跃进<sup>1)</sup> 陈火旺<sup>1)</sup>

<sup>1)</sup>(国防科学技术大学计算机学院 长沙 410073)

<sup>2)</sup>(北京航空航天大学计算机学院 北京 100083)

<sup>3)</sup>(德雷塞尔大学信息科学与技术学院 费城 美国)

**摘 要** 数据流具有数据持续到达、到达速度快、数据规模巨大等特点,这些都给数据流挖掘领域的研究工作带来了新挑战,而其中分类算法更是当前的研究热点。Domingos 等在 VFDT 中利用 Hoeffding 不等式很好地解决了在数据流上进行单遍扫描获取高精度决策树的问题。Gama 等对 VFDT 进行扩展并实现了 VFDTc,使系统能够处理连续属性。Peng 等在传统数据挖掘环境下提出了基于模糊理论的连续属性平滑离散化方法。基于前述工作,作者设计并实现了一种基于线索化排序二叉树的增量模糊决策树分类算法 fVFDT,其主要贡献有如下 4 点:(1)第一次设计并实现了数据流上的基于线索化二叉排序树(TBST)的连续属性处理方法。相比 VFDT,fVFDT 的样本插入时间复杂度由  $O(n^2)$  降低到  $O(n \log n)$ 。当新样本到达时,VFDTc 需要更新  $O(\log n)$  个属性节点,而 fVFDT 只需要更新相应的一个节点即可;(2)改进了 VFDTc 连续属性的最佳划分节点选取的计算方法,使其时间复杂度由  $O(n \log n)$  降低到  $O(n)$ ;(3)根据 Fayyad 等的研究成果,相比 VFDTc,fVFDT 只需从更少的备选划分节点中选取最佳节点,备选划分节点数由  $O(n)$  降低到  $O(\log n)$ ;(4)改进了传统数据挖掘环境下的基于模糊理论的连续属性平滑离散化方法,有效地处理了噪声数据,很好地提高了分类精度。

**关键词** 数据流;线索化二叉排序树;连续属性;模糊离散化;增量;VFDT

中图法分类号 TP181

## An Incremental Fuzzy Decision Tree Classification Method for Data Streams Mining Based on Threaded Binary Search Trees

WANG Tao<sup>1)</sup> LI Zhou-Jun<sup>2)</sup> HU Xiao-Hua<sup>3)</sup> YAN Yue-Jin<sup>1)</sup> CHEN Huo-Wang<sup>1)</sup>

<sup>1)</sup>(School of Computer Science, National University of Defense Technology, Changsha 410073)

<sup>2)</sup>(School of Computer Science & Engineering, Beihang University, Beijing 100083)

<sup>3)</sup>(College of Information Science and Technology, Drexel University, Philadelphia PA, USA)

**Abstract** Decision tree classification is a well-studied problem in data mining. Recently, there has been much interest in mining data streams. Domingos and Hulten have presented a one-pass algorithm. Their system, VFDT, uses Hoeffding inequality to achieve a probabilistic bound on the accuracy of the tree constructed. Gama et al. have extended VFDT in two directions. Their system VFDTc can deal with continuous data and use more powerful classification techniques at tree leaves. Peng et al. present soft discretization method to solve continuous attributes in data mining. This paper revisits this problem and implemented a system fVFDT on top of VFDT and VFDTc. It has the following four contributions: (1) It presents a threaded binary search trees (TBST) approach for efficiently handling continuous attributes. It builds a threaded binary

收稿日期:2007-03-05;修改稿收到日期:2007-05-31。本课题得到国家自然科学基金(60573057)资助。王 涛,男,1976 年生,博士研究生,主要研究方向为数据流分类技术。E-mail: InsistStar@nudt.edu.cn。李舟军,男,1963 年生,教授,博士生导师,主要研究领域为安全协议形式化分析、进程代数理论、数据挖掘。胡小华,男,1965 年生,博士,教授,主要研究领域为生物信息学、数据挖掘、文本挖掘、粗糙集理论等。颜跃进,男,1976 年生,博士,讲师,主要研究方向为数据挖掘和 OLAP。陈火旺,男,1936 年生,博士生导师,中国工程院院士,主要研究领域为软件理论和软件工程。

search tree, and its processing time for values inserting is  $O(n \log n)$ , while VFDT's processing time is  $O(n^2)$ . When a new example arrives, VFDTc need update  $O(\log n)$  attribute tree nodes, but fVFDT just need update one necessary node. (2) It improves the method of getting the best split-test point of a given continuous attribute. Comparing to the method used in VFDTc, it improves from  $O(n \log n)$  to  $O(n)$  in processing time. (3) Comparing to VFDTc, fVFDT's candidate split-test number decrease from  $O(n)$  to  $O(\log n)$ . (4) It uses soft discretization method in data streams mining to solve the problem of noise data.

**Keywords** data streams; threaded binary search tree; continuous attribute; soft discretization; incremental; VFDT

1 引言

作为统计学、人工智能等学科的交叉学科,数据挖掘近年来正逐渐成为研究热点,各种数据挖掘技术相继被提出并广泛运用,以达到从大量复杂资料中获取有用信息的目的.随着信息的大量产生,需要处理的数据正以每天数以百万计甚至没有上限的速度增长,如何从这些连续不断的数据流(data streams)中挖掘有用的信息,目前已成为我们面临的一大重要挑战<sup>[1-4]</sup>.

令  $t$  表示任一时间戳,  $a_t$  表示在该时间戳到达的数据向量,数据流可以表示成  $\{\cdots, a_{t-1}, a_t, a_{t+1}, \cdots\}$ . 区别于传统的数据模型,数据流模型具有以下 3 个特性:(1) 数据高速到达,实时性要求高;(2) 数据规模宏大,不可能把所有的数据都放入内存甚至是硬盘;(3) 数据一经处理,除非特意保存,否则不能被再次取出处理,或者再次提取数据代价昂贵.传统的数据挖掘方法必须将数据全部存储到介质中,然后通过访问存储介质进行挖掘.但由于数据的快速到达和数据规模巨大的原因,传统数据挖掘技术难以满足数据流挖掘的要求.

分类是一种非常重要的数据挖掘技术,其目的是根据已有的数据集学习构造一个分类函数或分类模型,该分类模型能够将新到样本映射到一个具体的类别上.传统的分类模型包括决策树、决策规则、贝叶斯理论分类、反向传播法、关联分类法、K 近邻分类器、SVM、范例式推理、进化算法、粗糙集法及模糊集合法等.其中决策树模型是最普遍的一种分类模型,它具有很好的可理解性.数据流挖掘的分类方法比传统的分类在实时性和存储限制等方面面临更多的挑战,同时在诸如电子邮件的区分、个性化网

站、电脑入侵检测等方面也有着更好的应用<sup>[5]</sup>.

Domingos 和 Hulten<sup>[6]</sup>的 VFDT 研究了如何在数据流上构造决策树的问题.他们的算法能够以一定的概率保证所构造决策树的精度. Gama 等<sup>[7]</sup>设计实现了 VFDTc,对 VFDT 从两个方向进行了扩展:处理连续属性的能力和在叶节点上使用分类能力更强的贝叶斯分类技术.

Peng 等<sup>[8]</sup>研究了传统数据挖掘环境下基于模糊理论的连续属性平滑离散化方法.该方法提高了决策树的抗噪声能力,从而提高了分类精度.

2 相关研究工作

2.1 VFDT

VFDT(Very Fast Decision Tree)<sup>[6]</sup>是一种基于 Hoeffding 不等式针对数据流挖掘环境建立分类决策树的方法,它通过不断地将叶节点替换为分支节点而生成.其最主要的创新是利用 Hoeffding 不等式<sup>[9]</sup>确定叶节点变为分支节点所需要的样本数目.

VFDT 最初并没有介绍有关连续属性的处理方法,在其后继研究中才加以介绍.对于连续属性,当 VFDT 新建一个叶节点的时候,为每个连续属性从最先到达的样本中选取并保存  $M$  个不同的取值.这些取值在样本到达的时候就已经通过排序数组被排序,并且每两个不同分类的相邻取值的中间节点作为备选划分节点被维护.一旦某个连续属性已经有了  $M$  个不同取值,则不再增加备选划分节点,只是把新到样本用于评价现有备选划分节点.根据叶节点在树中所处层的不同以及当前可用内存大小的不同,每个叶节点使用不同的  $M$  值.

2.2 VFDTc

VFDTc<sup>[7]</sup>从两个方面对 VFDT 进行了扩展:处

理连续属性和在叶节点应用贝叶斯分类器.

现实中的数据大都含有连续属性,传统的决策树学习方法需要对连续属性值排序,而排序操作是非常耗时的,显然不适合数据流的实时性要求. VFDTc 针对该问题提出了一个有效的基于属性树的解决方法.

VFDTc 中每个连续属性划分时都分为两个分枝,划分测试形如  $attr_i \leq cut\_point$ ,左右两个分枝分别对应划分测试的真和假,其中  $cut\_point$  为属性的所有可能取值. 为了比较各个划分的优劣,需要根据数值  $n_{ijk}$  计算样本取值小于等于和大于  $cut\_point$  的分布数目. VFDTc 在决策树的每个叶节点上保存到达该节点的样本的类分布向量. 对于每个连续属性  $j$ ,叶节点都保存一个二叉树结构(属性树). 该二叉属性树的每个节点都对应属性  $j$  的一个取值  $i$ ,同时,在每个树节点上维持两个向量  $\mathbf{VE}$  和  $\mathbf{VH}$ (维度为  $k$ ),分别保存  $\leq i$  和  $> i$  的样本数目. 当样本到达节点后,该节点的所有连续属性二叉树都要进行更新. 新到样本的插入算法的时间耗费是  $O(n \log n)$ ,其中  $n$  表示属性  $j$  所观测到的不同取值的数目.

VFDTc 提出了处理连续属性的方法,但是它在划分节点的选择上把所有连续属性的可能取值都作为备选划分,这带来了很大的开销. Fayyad 等<sup>[10]</sup>已经证明连续属性的两个不同分类的紧邻值之间的中间值才可能是最佳划分节点,这将大大减少备选划分节点的数目,这在 VFDTc 中并没有得到应用,因此需要加以改进.

2.3 连续属性平滑离散化方法

Peng 等<sup>[8]</sup>基于模糊理论提出了传统数据挖掘环境下的连续属性平滑离散化方法.

针对连续属性,决策树分类方法大都采用将连续属性通过划分节点将其离散化为两个或多个离散区间的方法进行处理. 这些称为陡峭离散化的方法在分类界限清晰的情况下得到了很好的应用,但现实数据存在大量非确定信息及噪声数据,这导致很多情况下分类界限并不清晰.

Peng 等<sup>[8]</sup>基于模糊理论,将连续属性的划分节点模糊化,提出了平滑离散化的模糊决策树方法. 该方法利用模糊决策树的非确定性解决了噪声数据等问题,提高了分类精度.

3 基本定义

定义 1. 数据流. 令  $t$  表示任一时间戳,  $a_t$  表示

在该时间戳到达的数据向量,数据流可以表示成  $\{\cdots, a_{t-1}, a_t, a_{t+1}, \cdots\}$ .

定义 2. 决策树. 决策树是一个类似于流程图的树结构,其中每个内部节点表示在一个属性上的划分,每个树分枝代表一个划分输出,每个树叶节点表示类别或类别分布.

定义 3. 信息增益. 设  $S$  是  $s$  个数据样本的集合. 假定类标号属性具有  $m$  个不同值.  $s_i$  是类别为  $C_i$  的样本数,则样本集合的熵或期望信息为

$$I(s_1, s_2, \cdots, s_m) = - \sum_{i=1}^m p_i \log_2(p_i).$$

设属性  $A$  具有  $v$  个不同取值  $\{a_1, a_2, \cdots, a_v\}$ ; 可以用属性  $A$  将  $S$  划分为  $v$  个子集  $\{S_1, S_2, \cdots, S_v\}$ , 其中  $S_j$  为  $S$  中属性  $A$  取值为  $a_j$  的样本子集;  $s_{ij}$  是子集  $S_j$  中类别为  $C_i$  的样本数目,则由属性  $A$  划分成多区间的熵或期望信息为

$$E(A) = \sum_{i=1}^v \frac{s_{1j} + \cdots + s_{mj}}{s} I(s_{1j}, \cdots, s_{mj}),$$

其中项  $\frac{s_{1j} + \cdots + s_{mj}}{s}$  为子集  $S_j$  的权,等于子集  $S_j$  的样本数除以  $S$  中的样本总数. 给定子集  $S_j$ ,

$$I(s_{1j} + \cdots + s_{mj}) = - \sum_{i=1}^m p_{ij} \log_2(p_{ij}),$$

其中  $p_{ij} = \frac{|s_{ij}|}{|s_j|}$  为  $S_j$  中样本属于类  $C_i$  的概率. 则由属性  $A$  进行划分的信息增益为  $Gain(A) = I(s_1, s_2, \cdots, s_m) - E(A)$ .

定义 4. 划分节点. 在决策树的构造过程中,会根据属性划分的信息增益利用阈值  $T$  将连续属性划分为两个或多个离散区间,阈值  $T$  称为连续属性的划分节点.

定义 5. Hoeffding 不等式. Hoeffding 不等式是对误差概率的一种严格的理论限制. 设  $\{X_i\}_{i=1}^m$  是  $m$  个随机变量,  $0 \leq X_i \leq r$ , 令  $\bar{X} = \frac{1}{m} \sum_{i=1}^m X_i$ ,  $\bar{X}$  的数学期望为  $\mu$ , 对于给定  $\epsilon > 0$ , Hoeffding 不等式形如:  $Pr(|\bar{X} - \mu| \geq \epsilon) \leq 2^{-2m\epsilon^2/r^2}$ .

定义 6. Hoeffding bound. 假设变量  $r$  取值范围为  $R$ , 观测  $n$  个样本后,样本观测平均值为  $\bar{r}$ , 则 Hoeffding bound 保证样本真值以置信度  $1 - \delta$  落于  $\bar{r} \pm \epsilon$  区间范围内,其中  $\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$ .

定义 7. 平滑离散化. 陡峭离散化是通过一个划分节点(阈值)将连续属性分为几个彼此不相交的离散

区间. 平滑离散化是利用  $\Omega$  上的满足  $\sum_{r=1}^k A_r(a) = 1$ ,  $\forall a \in \Omega$  的模糊划分  $Q = \{A_1, A_2, \dots, A_k\}$  将连续属性划分为几个边界重叠的区间. 平滑离散化由三个参数确定, 第一个是交叉点  $T$ , 另外两个是满足  $A_1(a) + A_2(a) = 1$  的模糊集合  $A_1$  和  $A_2$  的隶属度函数.

**定义 8.** 模糊信息增益. 样本集合  $S$  的模糊熵为

$$E_F(S) = - \sum_{i=1}^m p(C_i, S) \log p(C_i, S),$$

其中  $p(C_i, S) = \sum_{a_j \in C_i} (A_1(a_j) + A_2(a_j))$ . 由属性  $A$  进行平滑离散化划分成多区间的模糊熵或期望信息为

$$E_F(A) = \frac{N_F^{S_1}}{N_F^S} E_F(S_1) + \frac{N_F^{S_2}}{N_F^S} E_F(S_2).$$

其中,

$$E_F(S_1) = - \sum_{i=1}^m p(C_i, S_1) \log p(C_i, S_1),$$

$$E_F(S_2) = - \sum_{i=1}^m p(C_i, S_2) \log p(C_i, S_2),$$

$$p(C_i, S_k) = N_F^{S_k C_i} / N_F^{S_k}, \quad k=1, 2,$$

$$N_F^S = \sum_{j=1}^{|S|} (A_1(a_j) + A_2(a_j)),$$

$$N_F^{S_1} = \sum_{j=1}^{|S|} A_1(a_j), \quad N_F^{S_2} = \sum_{j=1}^{|S|} A_2(a_j),$$

$$N_F^{S_k C_i} = \sum_{a_j \in C_i} A_k(a_j), \quad k=1, 2,$$

模糊信息增益  $Gain_F(A) = E_F(S) - E_F(A)$ .

## 4 fVFDT 设计与技术细节

基于 VFDT 和 VFDTc, 改进平滑离散化技术, 我们设计并实现了一个名为 fVFDT 的系统. 该系统利用线索化二叉排序树维持连续属性, 并且使用了更高效的最佳划分节点选取方法, 大大减少了算法执行时间; 改进平滑离散化方法, 同线索化二叉排序树结构相结合构造增量模糊决策树, 有效地解决了噪声数据问题, 提高了分类精度.

### 4.1 线索化二叉排序树结构

对于每个连续属性  $i$ , fVFDT 维持一个线索化二叉排序树. 树中的每个节点包括 *keyValue*, *classTotals*[ $k$ ], *left* 及 *right* 指针, *prev* 及 *next* 指针等几个属性. 其中 *keyValue* 用于记录到达样本的

属性  $i$ ; 向量 *classTotals*[ $k$ ] 记录属性  $i$  取值为 *keyValue* 类别为  $k$  的样本的数目; *left* 和 *right* 指针分别用于记录节点的左右孩子(基于  $\leq \text{keyValue}$  的判断); *prev* 和 *next* 指针分别用于记录节点的前驱和后继.

此外, fVFDT 还为每个连续属性维持一个头指针 *head* 用于遍历整个属性树.

### 4.2 新样本到达时的属性树更新过程

数据流分类算法构造决策树的一个很重要的问题就是用于保存信息以获取最佳划分节点的开销非常大. 离散属性的属性取值一般都不会太大, 因此其类别信息保存开销不会太高, 同样备选划分节点也不会太多.

对于取值很多的连续属性, 系统开销将会非常大. Domingos 和 Hulten<sup>[6]</sup> 提出了解决数据流上离散属性的方法, 并且在其后续工作中利用排序数组解决了连续属性问题. 但排序数组不管是对新样本的插入, 还是对利用排序数组计算最佳划分节点, 开销都是非常大的.

在 fVFDT 中, 每个 Hoeffding 树节点在变为叶节点前我们都在该节点上为每个连续属性维持一个线索化二叉排序树.

当有新样本  $(x, k)$  到达时, 连续属性  $i$  所对应的属性树的更新过程如图 1 所示. VFDT 的新样本插入的时间复杂度为  $O(n^2)$ , 而 fVFDT 的时间复杂度为  $O(n \log n)$  (其中  $n$  为当前节点上所观察到的连续属性  $i$  的不同取值的数目).

```

Procedure InsertValuefTBSTree( $x, k, fTBSTree$ )
Begin
  While( $fTBSTree \rightarrow right \neq \text{NULL} \parallel fTBSTree \rightarrow left \neq \text{NULL}$ )
  If ( $fTBSTree \rightarrow keyValue == x$ ) then break;
  elseIf ( $fTBSTree \rightarrow keyValue > x$ ) then
     $fTBSTree = fTBSTree \rightarrow left$ ;
  else  $fTBSTree = fTBSTree \rightarrow right$ ;
  Creates a new node  $curr$  based on  $x$  and  $k$ ;
  If( $fTBSTree.keyValue == x$ ) then
     $fTBSTree.classTotals[k]++$ ;
  elseIf ( $fTBSTree.keyValue > x$ ) then  $fTBSTree.left = curr$ ;
  else  $fTBSTree.right = curr$ ;
  Threads the tree;
End

```

图 1 类别  $k$  取值  $x$  的样本插入属性树的过程

新样本到达时, VFDTc 需要更新  $O(\log n)$  个属性树节点, 而 fVFDT 只需要更新一个节点就可以了.

4.3 新样本到达时的属性树线索化过程

新样本到达时, fVFDT 需要线索化已有二叉排序属性树. 如果新样本的取值和属性树中已有节点的取值相同, 只需要修改相应的类统计信息, 不需要重新线索化, 否则属性树需要重新线索化.

当新样本到达, 属性树需要更新线索化时, 最多只需要更新三个节点的指针信息, 并且更新过程的时间复杂度是  $O(1)$  的. 该线索化更新过程是嵌入到新到样本时的属性树更新过程  $InsertValueTBSTree(x, k, TBSTree)$  中的, 因此 fVFDT 中引入的线索化机制并不会增加新样本插入的时间复杂度, 依然为  $O(n \log n)$ .

4.4 连续属性的平滑离散化划分过程

利用线索化二叉排序树的特性, 我们使用了一个更高效的最佳划分节点选取的方法.

假设某个决策树节点含有  $N$  个样本, 连续属性  $i$  的样本不同取值为  $a_1, a_2, \dots, a_n$ , 系统将为属性  $i$  维护一个线索化二叉排序属性树. 所有相邻取值的中间节点值  $T = (a_i + a_{i+1}) / 2$  都作为该属性的备选划分节点. 为计算划分节点的模糊信息增益, 需要知道样本取值  $attr_i \leq T$  和  $attr_i > T$  的类分布. fVFDT 中决策树节点的属性值  $TBSTree.classTotals[k]$  用于计算模糊信息增益.

如图 2 所示, 根据模糊信息增益的计算公式, 从  $head$  头节点开始按照线索化顺序遍历整个属性树就可以计算所有备选划分节点的模糊信息增益, 从而选取该连续属性的最佳划分节点.

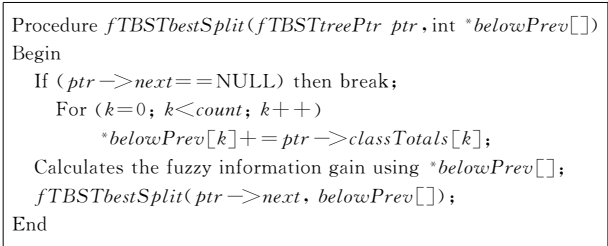


图 2 连续属性最佳划分节点选取算法

采用文献[11]中介绍的三角模隶属度函数计算方法, fVFDT 根据每个连续属性所记录的属性最大值、最小值和样本数实现连续属性的平滑离散化.

VFDTc 选取最佳划分节点的时间复杂度为  $O(n \log n)$ , 而我们 fVFDT 的时间复杂度仅为  $O(n)$  (这里  $n$  代表连续属性目前所观察到的不同取值的数目).

4.5 新样本分类过程

对于新样本, VFDT 从根节点开始, 在每个分支节点上进行测试, 完成从上到下的遍历过程, 最终的叶节点就是该样本的分类.

fVFDT 采用模糊决策树的 T-S 模分类方法. 为分类一个新样本, 首先利用  $T$  算子(模糊乘  $\otimes$ )计算出所有对某个分类隶属度非零的叶节点; 然后利用  $S$  算子(模糊加  $\oplus$ )计算出该样本对所有分类的隶属度; 最后, 利用去模糊化方法确定该样本的最终分类<sup>[12]</sup>.

5 实验结果

为验证我们所提出的基于线索化二叉排序属性树的模糊增量决策树算法将大大降低数据流挖掘的分类时间复杂度和提高抗噪声数据能力. 我们设计了三组实验, 分别用于验证算法的执行时间、分类错误率和决策树大小.

实验用机器配置为: 奔 IV/2GHz 的 CPU, 内存大小是 512MB, 操作系统为 Linux RedHat9.0. 实验采用和文献[6]中相同的数据环境, 都是用名为 treeData 的工具生成的数据流.

5.1 执行时间比较

算法的时间复杂度理论分析比较如表 1 所示.

表 1 算法执行时间

算法	样本插入时间 复杂度	信息增益计算时间 复杂度	备选划分节点数 复杂度
VFDT	$O(n^2)$	$O(n)$	$O(\log n)$
VFDTc	$O(n \log n)$	$O(n \log n)$	$O(n)$
fVFDT	$O(n \log n)$	$O(n)$	$O(\log n)$

VFDTc 的主要目的是为了说明在决策树叶节点上使用更强的分类技术将提高分类器的性能. 而贝叶斯分类器的引入将会大大增加系统的处理时间, 因此 VFDTc 的执行时间会有所增加. 为了和 VFDTc 中所提到的连续属性处理方法进行处理时间的比较, 我们根据文献[7]中所提供的算法实现了其连续属性处理部分.

表 2 是三个算法的执行时间实验比较结果. 在该实验中, 所用的实验数据为 treeData 产生, 为了更好地比较算法对连续属性的处理能力, 本实验所采用的数据为 20 个连续属性, 没有离散属性, 样本数目为  $10^7$  个. 10 次实验取平均值, 结果显示: fVFDT 比 VFDT 平均执行时间减少 16.66%, fVFDT 比 VFDTc 平均执行时间减少 6.25%.

表 2 算法执行时间实验结果

样本数	算法执行时间/s		
	VFDT	VFDTc	fVFDT
10000	4.66	4.21	3.75
20736	9.96	8.83	8.12
42996	22.88	20.59	18.57
89156	48.51	43.57	40.87
184872	103.61	93.25	87.12
383349	215.83	187.77	175.23
794911	522.69	475.65	441.61
1648326	1123.51	1022.39	939.35
3417968	2090.31	1839.45	1758.89
7087498	3392.94	3053.65	2882.23
14696636	5209.47	4688.53	4389.35
30474845	8203.05	7382.75	6850.12
43883922	13431.02	11953.61	11068.23
90997707	17593.46	15834.12	15020.46
100000000	18902.06	16822.86	15986.23

5.2 分类错误率比较

文献[8]中实验证明了平滑离散化的方法将有效解决噪声数据问题,从而提高分类精度.通过和线索化二叉排序属性树的有效结合,平滑离散化方法很好地应用在数据流环境下,提高了分类精度.如图 3 所示,在 10% 噪声数据情况下,VFDT 的分类错误率逼近于 12.5%,而 fVFDT 的分类错误率逼近于 8%.实验很好地验证了平滑离散化方法对于分类精度的提高作用.

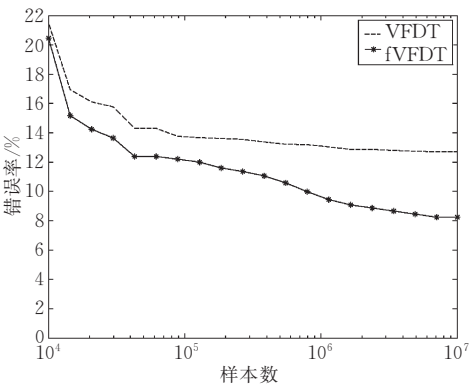


图 3 VFDT 和 fVFDT 在 10% 噪声数据下的精度比较

5.3 决策树大小比较

fVFDT 中针对连续属性的基于线索化排序二叉树的平滑离散化处理方法,并没有改变决策树的生成框架,只是采用新的数据结构提高了分类速度和分类精度,因此并没有改变决策树大小(决策树节点数目).

6 结束语

本文在 VFDT 和 VFDTc 的基础上,改进平滑

离散化方法,设计并实现了数据流环境下的基于线索化排序二叉属性树的增量模糊决策树算法 fVFDT.针对连续属性的处理问题,我们设计并实现了线索化二叉排序属性树的新方法.该方法很好地降低了新样本插入和最佳划分节点选取的时间复杂度.针对样本处理最耗时的部分,同 VFDT 相比,其新样本插入的时间复杂度由  $O(n^2)$  降低为  $O(n \log n)$ ;对于划分节点信息增益的计算,同 VFDTc 相比,其时间复杂度由  $O(n \log n)$  降低为  $O(n)$ ;利用 Fayyad<sup>[10]</sup>的结论,同 VFDTc 相比,备选划分节点的数目由  $O(n)$ 降低为  $O(\log n)$ .针对噪声数据问题,通过和线索化二叉排序属性树的有效结合,平滑离散化方法很好地应用在数据流环境下,提高了分类精度

fVFDT 中并没有考虑概念漂移问题<sup>[5,13-15]</sup>, CVFDT<sup>[15]</sup>已经提供了解决概念漂移的方法,能否将当前方法推广到存在概念漂移的情形,是我们下一步的研究重点.

参 考 文 献

[1] Babcock B, Babu S, Datar M, Motawani R, Widom J. Models and issues in data stream systems//Proceedings of the PODS. 2002

[2] Jin R, Agrawal G. Efficient decision tree construction on streaming data//Proceedings of the ACM SIGKDD 2003. 2003; 571-576

[3] Last M. Online classification of nonstationary data streams. Intelligent Data Analysis, 2002, 6(2): 129-147

[4] Muthukrishnan S. Data streams: Algorithms and applications//Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, 2003

[5] Xie Q H. An efficient approach for mining concept-drifting data streams [ M. S. dissertation]. National University of Tainnan, Tainan, China, 2004

[6] Domingos P, Hulten G. Mining high-speed data streams//Proceedings of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining. 2000; 71-80

[7] Gama J, Rocha R, Medas P. Accurate decision trees for mining high-speed data streams//Domingos P, Faloutsos C eds. Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining. ACM Press, 2003; 523-528

[8] Peng Y H, Flach P A. Soft discretization to enhance the continuous decision tree induction//Proceedings of the ECML/PKDD' 2001 Workshop IDDM' 2001. Freiburg, Germany, 2001

- [9] Hoeffding W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963, 58: 13-30
- [10] Fayyad U M, Irani K B. On the handling of continuous-valued attributes in decision tree generation on learning. *Machine Learning*, 1992, 9: 87-102
- [11] Zeidler J, Schlosser M. Continuous-valued attributes in fuzzy decision trees//*Proceedings of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. 1996: 395-400
- [12] Guevova M, Holldobter, Storr H-P. Incremental fuzzy decision trees//*Proceedings of the 25th German Conference on Artificial Intelligence (KI2002)*, 2002: 61-67
- [13] Fan Wei. StreamMiner: A classifier ensemble-based engine to mine concept drifting data streams//*Proceedings of the VLDB'2004*. 2004: 1257-1260
- [14] Wang H, Fan W, Yu P, Han J. Mining concept-drifting data streams using ensemble classifiers//*Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. Washington DC, USA, 2003: 226-235
- [15] Kelly M G, Hand D J, Adams N M. The impact of changing populations on classifier performance//*Proceedings of the KDD'99*, 1999: 367-371
- [16] Hulten G, Spencer L, Domingos P. Mining time-changing data streams//*Proceedings of the ACM SIGKDD 2001*. 2001: 97-106
- [17] Cezary, Janikow Z. Fuzzy decision trees: Issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 1998, 28(1): 1-14
- [18] Quinlan J R. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993



**WANG Tao**, born in 1976, Ph. D. candidate. His research interests include data streams mining and fuzzy methodologies.

**LI Zhou-Jun**, born in 1963, professor, Ph. D. supervisor. His research interests include concurrency theory and process algebra, formal analysis and verification of security protocols, data mining and bioinformatics, etc.

**HU Xiao-Hua**, born in 1965, professor. His research interests include biomedical literature, data mining, bioinformatics, text mining, semantic web mining and reasoning, rough set theory and application, information extraction and information retrieval.

**YAN Yue-Jin**, born in 1976, Ph. D.. His research interests include data mining and OLAP.

**CHEN Huo-Wang**, born in 1936, professor, Ph. D. supervisor, member of Chinese Academy of Engineering. His research interests include software theory and soft engineering.

## Background

This work is supported in part by the National Natural Science Foundation of China under grant No. 60573057 (Research on Some Key Algorithms in Data Mining).

VFDT is an anytime system that builds decision trees using constant memory and constant time per example. It uses Hoeffding bounds to guarantee that its output is asymptotically nearly identical to that of conventional learner.

VFDTc extends VFDT in two directions: The ability to deal with continuous data and the use of more powerful classification techniques at tree leaves. It's most relevant property is the ability to obtain a performance similar to a standard decision tree even for medium size datasets.

In this paper, the authors present a new system fVFDT on top of VFDT and VFDTc.