DOI:10.7544/issn1000-1239.2017.20160154
54(7):1477-1487,2017

一种基于决策森林的单调分类方法

许 行[1] 王文剑[1,2] 任丽芳[1,3]
[1](山西大学计算机与信息技术学院 太原 030006)
[2](计算智能与中文信息处理教育部重点实验室(山西大学) 太原 030006)
[3](山西财经大学应用数学学院 太原 030006)
(xuh102@126.com)

A Method for Monotonic Classification Based on Decision Forest

Xu Hang[1], Wang Wenjian[1,2], and Ren Lifang[1,3]
[1](*School of Computer and Information Technology, Shanxi University, Taiyuan* 030006)
[2](*Key Laboratory of Computational Intelligence and Chinese Information Processing (Shanxi University), Ministry of Education, Taiyuan* 030006)
[3](*School of Applied Mathematics, Shanxi University of Finance and Economics, Taiyuan* 030006)

Abstract Monotonic classification is an ordinal classification problem in which the monotonic constraint exists between features and class. There have been some methods which can deal with the monotonic classification problem on the nominal datasets well. But for the monotonic classification problems on the numeric datasets, the classification accuracies and running efficiencies of the existing methods are limited. In this paper, a monotonic classification method based on decision forest (MCDF) is proposed. A sampling strategy is designed to generate decision trees, which can make the sampled training data subsets having a consistent distribution with the original training dataset, and the influence of non-monotonic noise data is avoided by the sample weights. It can effectively improve the running efficiency while maintaining the high classification performance. In addition, this strategy can also determine the number of trees in decision forest automatically. A solution for the classification conflicts of different trees is also provided when the decision forest determines the class of a sample. The proposed method can deal with not only the nominal data, but also the numeric data. The experimental results on artificial, UCI and real datasets demonstrate that the proposed method can improve the monotonic classification performance and running efficiency, and reduce the length of classification rules and solve the monotonic classification problem on large datasets.

Key words monotonic classification; decision tree; monotonic consistency; decision forest; ensemble learning

摘 要 单调分类问题是特征与类别之间带有单调性约束的有序分类问题.对于符号数据的单调分类问题已有较好的方法,但对于数值数据,现有的方法分类精度和运行效率有限.提出一种基于决策森林的单调分类方法(monotonic classification method based on decision forest, MCDF),设计采样策略来构

收稿日期:2016-03-11;**修回日期**:2016-08-08
基金项目:国家自然科学基金项目(61673249,61503229);山西省回国留学人员科研基金项目(2016-004);山西省研究生教育创新项目(2016BY003)
This work was supported by the National Natural Science Foundation of China (61673249, 61503229), the Scientific Research Foundation for Returned Scholars of Shanxi Province (2016-004), and the Innovation Project of Shanxi Graduate Education (2016BY003).
通信作者:王文剑(wjwang@sxu.edu.cn)

造决策树,可以保持数据子集与原数据集分布一致,并通过样本权重避免非单调数据的影响,在保持较高分类精度的同时有效提高了运行效率,同时这种策略可以自动确定决策森林中决策树的个数.在决策森林进行分类时,给出了决策冲突时的解决方法.提出的方法既可以处理符号数据,也可以处理数值数据.在人造数据集、UCI及真实数据集上的实验数据表明:该方法可以提高单调分类性能和运行效率,缩短分类规则的长度,解决数据集规模较大的单调分类问题.

关键词 单调分类;决策树;单调一致性;决策森林;集成学习

中图法分类号 TP181

单调分类是一种特殊的分类问题,广泛存在于实际生活中,如大学综合水平的评定问题,其中"科研水平"、"师资队伍"、"教学质量"是3个重要指标,其得分很显然就是一种序关系,大学综合水平的"评定等级"若为"高、中、低",则它们之间也具有优劣次序,这一问题的特征(科研水平、师资队伍、教学质量)与类别(评定等级)之间存在以下单调性约束:当一所学校在科研水平、师资队伍、教学质量这3个特征上的分数都不比另一所学校低时,它的评定等级就一定不会比另一所学校低.此外,还有诸如决策风险分析、企业绩效考核、消费者满意度评价等问题都具有同样的特点,这类问题就是单调分类问题.与一般分类问题不同,单调分类问题的特征与类别上的取值都存在序的关系,且2者之间满足单调性约束.

对于一般的分类问题,已经有很多分类方法,如决策树、神经网络、支持向量机等,这些方法一般不考虑特征与类别上的序关系,所以不能得到单调一致的分类规则.目前对于单调分类问题已有一些研究,如以粗糙集为理论框架的模型解决单调分类问题[1-6],但这些方法获得的单调一致的分类规则数量有限,且只能用于符号数据.对于数值数据的单调分类问题,我国学者Hu等人[7-10]提出的单调决策树是目前较好的方法,他们在一定程度上解决了单调性和泛化能力之间的冲突,还将序的关系引入经典Shannon熵,提出有序信息熵概念,并在此基础上设计了基于有序信息熵的决策树算法(rank entropy based monotonic decision tree, REMT).这种方法能够产生结构简单、易于理解的规则集,但得到的精度相对有限,且存在过度拟合问题.

决策森林源于随机森林[11],是以决策树为基础分类器的集成算法,可以有效避免过度拟合,且能够提高分类精度,因而受到了研究者的广泛关注.但决策森林目前只用于解决一般的分类问题,对于单调分类问题,构造决策森林时面临3种困难:1)单个决策树的构造需要在不同的训练子集上进行,如何保证子集的单调性;2)如何确定决策森林的规模;3)集成规则出现冲突时如何解决.

本文提出一种用于解决单调分类问题的基于决策森林的单调分类方法,设计采样策略来构造决策树,该策略考虑数据集单调一致的特点,可避免非单调数据的噪声影响,同时这种策略可以自动确定决策森林中决策树的个数.在决策森林进行分类时,本文还提出叶子节点上的规则支持度的概念,从而给出决策冲突时的解决方法.本文提出的方法既可以处理符号数据,也可以处理数值数据.

1 背景知识

1.1 单调分类问题

令 $U=\{x_1,x_2,\cdots,x_n\}$ 为数据集,$A=\{a_1,a_2,\cdots,a_m\}$ 是描述数据的特征集,x_i 在特征 a 上的值记为 $a(x_i)$,x_i 的类标签记为 $y(x_i)$.如果 $y(x_i)$ 的值域 $\{c_1,c_2,\cdots,c_s\}$ 上存在序关系 $c_1<c_2<\cdots<c_s$,则称此问题是一个有序分类问题[12].

对于一个有序分类问题,如果 $\forall x_i,x_j\in U$ 在每一个特征 $a_k\in A$ 上都满足若 $a_k(x_i)\leqslant a_k(x_j)$,则 $y(x_i)\leqslant y(x_j)$,那么该问题为单调分类问题[12].

给定 $x_i\in U$,若特征集 $B\subseteq A$,则在特征集 B 上 x_i 的优势类和劣势类分别定义为

$$[x_i]_B^{\geqslant}=\{x\in U\mid a_k(x)\geqslant a_k(x_i),\ \forall a_k\in B\};\quad(1)$$

$$[x_i]_B^{\leqslant}=\{x\in U\mid a_k(x)\leqslant a_k(x_i),\ \forall a_k\in B\}.\quad(2)$$

1.2 决策森林

决策森林是由多个单棵决策树组成集成分类器的集成学习方法,可表示为 $\{h(x,\theta_k),k=1,2,\cdots\}$,其中,$x$ 是输入样本,θ_k 是独立同分布的随机变量,$h(x,\theta_k)$ 表示一棵决策树.对于输入样本 x,每棵决策树判断其类别后投票给与其判断结果相符的类,得票最多的一类为最终分类结果.构建决策森林的方式主要有基于数据集的构造方式和基于特征集的构造方式.

构建决策森林模型时主要考虑4方面:

1) 当采用基于数据集的构造方式构造决策森林时,训练数据子集如何构成;

2) 当采用基于特征集的构造方式构造决策森林时,特征子集如何构成;

3) 决策树的数目如何确定;

4) 生成的所有决策树如何集成为决策森林模型.

2　决策森林单调分类方法

2.1　单棵决策树的构造

决策森林的预测能力与每棵决策树相关,为用较少的决策树构造出精度较高的决策森林,决策树必须要有一定的差异性.本文采用基于数据集的构造方式构造决策森林,通过为每个决策树构造有差异性的训练子集来获得决策树之间的差异性.构造单棵决策树的训练子集时,不能采用经典的随机采样、bagging[13]、boosting[14]等方法,因为随机采样和bagging方法都不考虑数据的分布情况,boosting方法对噪声的敏感性使其更易于产生过拟合现象[15],并且这些方法也没有考虑数据的单调性.Liang等人[16]提出一种较好的重采样方法,使得采样得到的子集间有一定的差异度又能较大程度地覆盖原始数据集,且与原数据集分布相同,但该方法不考虑数据的序关系,且只能用于符号数据的处理.

本文借鉴文献[16],将以上方法引入单调分类问题,可以同时处理符号数据和数值数据.在构造单棵决策树时,为保持单调分布的一致性,为每个样本加权,在多次采样时权值不是固定不变的,而是随着已有决策树的错误率和错分样本的单调一致性进行调整,在调整中去掉不一致的样本.另外,本文中训练子集的规模不是由用户指定的固定值,而是根据数据集的特征自动确定的.

1) 训练子集规模的确定

对于数据集U,文献[16]给出的训练子集规模M为

$$M=\frac{Z^2\times\sigma^2}{E^2},\tag{3}$$

其中,Z为置信水平下的Z-统计值,可以计算得出,E为可以接受的误差,由用户指定,这2个变量取值已知,因此计算M的关键在于数据集上的标准差σ的计算.文献[16]给出了符号数据上σ的计算方法.

为同时处理符号数据和数值数据,本文设计的标准差σ为

$$\sigma=\sum_{k=1}^{|A|}\sigma_{a_k},\tag{4}$$

其中,σ_{a_k}为每个特征上的特征标准差,$|A|$为特征的个数.每个特征上的取值既可以是符号型,也可以是数值型.

对于数值型特征,先统一各个特征的取值范围,将所有样本的特征取值归一化:

$$a'_k(x_i)=\frac{a_k(x_i)-a_{k_{\min}}(x)}{a_{k_{\max}}(x)-a_{k_{\min}}(x)},\tag{5}$$

其中,$a_{k_{\min}}(x)$和$a_{k_{\max}}(x)$分别表示所有样本在特征a_k上的最小值和最大值,然后计算该特征的特征标准差:

$$\sigma_{a_k}=[(a'_k(x_1)-\bar{a}'_k(x))^2+(a'_k(x_2)-\bar{a}'_k(x))^2+\cdots+(a'_k(x_N)-\bar{a}'_k(x))^2]^{\frac{1}{2}}\left(\frac{1}{N}\right)^{-\frac{1}{2}},\tag{6}$$

其中,$\bar{a}'_k(x)$表示所有样本在特征a_k上归一化后的平均值.

对于符号型特征,与文献[16]不同,本文采用单个特征的不相似度而不是基于所有特征上的不相似度来计算特征标准差σ_{a_k}.

$$\sigma_{a_k}=\frac{\sum_{i=1}^{|U|}\sum_{j=1}^{|U|}\delta(a_k(x_i),a_k(x_j))}{|U|^2},\tag{7}$$

其中,$\delta(a_k(x_i),a_k(x_j))=\begin{cases}1, & a_k(x_i)\neq a_k(x_j).\\ 0, & a_k(x_i)=a_k(x_j).\end{cases}$

一般地,为保证学习器的训练效率,训练子集的规模M不宜太大,若M与样本总数N超过一定比率θ(如$\theta=5\%$),需要调整训练子集的规模,调整方法为

$$M=\left\lfloor\frac{MN}{(M+N)\times 10}\right\rfloor\times 10.\tag{8}$$

确定训练子集样本的规模算法总结如下.

算法1. 确定训练子集样本的规模算法.

输入:单调数据集U、参数Z,E,θ;

输出:训练子集的规模M.

Step1. 按式(4)计算数据集的标准差:若特征为数值型,则按式(5)将所有样本在该特征上的取值归一化,然后按式(6)计算其特征标准差;若特征为符号型,则按式(7)计算其特征标准差;

Step2. 按式(3)计算训练子集的规模M;

Step3. 如果$M>\theta|U|$,则按式(8)调整训练子集的规模;

Step4. 返回M.

2) 训练子集的采样

采样训练子集前,先计算原始训练集U中每个类别的样本占所有样本的比例p_l:

$$p_l=\frac{|C_l|}{|U|},\tag{9}$$

其中,$l=1,2,\cdots,s$,$C_l=\{x_i\in U\mid y(x_i)=c_l\}$,$i=1,2,\cdots,n$. 采样时要确保各个类别的样本数在原训练集与采样出的训练子集中的比例相同. 之后通过迭代采样训练子集,每次迭代产生 1 个训练子集. 迭代过程如下:

初始状态下,原训练集 U 中所有样本的权重相等,即每个样本的初始权重为

$$w(x_i)=\frac{1}{n}, \tag{10}$$

其中,n 为原训练集中样本的数量.

首先采样第 1 个训练子集 U_1,在原训练集 U 中随机选择 M 个样本,并保证其中每个类别的样本数与 M 的比例为 p_l,$l=1,2,\cdots,s$. 采样下一个训练子集 U_t 时,先在前一个训练子集上生成决策树,计算该决策树的加权误差 ε_t:

$$\varepsilon_t=\sum_{i=1}^{n}w(x_i)I(\hat{y}^t(x_i)\neq y(x_i)), \tag{11}$$

其中,t 为迭代次数,$\hat{y}^t(x_i)$ 表示第 t 次迭代中 x_i 的预测类,$y(x_i)$ 表示 x_i 的实际类,如果 $\hat{y}^t(x_i)\neq y(x_i)$,$I(\hat{y}^t(x_i)\neq y(x_i))=1$,否则,$I(\hat{y}^t(x_i)\neq y(x_i))=0$.

根据错误率更新原训练集的样本权重:对于错分的样本,首先判断该样本的单调一致性,本文用包含度描述样本的单调一致性. 定义 x_i 在特征集 A 与类标签 y 上的包含度为

$$D(x_i)=\frac{|[x_i]_y^{\geqslant}\cap[x_i]_A^{\geqslant}|}{|[x_i]_y^{\geqslant}|}, \tag{12}$$

$D(x_i)$ 的值越大,表示样本 x_i 在特征集与类别上取值的单调一致性越高. 这里把满足 $D(x_i)\geqslant e$ 的样本 x_i 看作单调一致的样本,否则为非单调一致的样本,e 为给定的参数.

对于错分的单调一致样本,增加其权重:

$$w_{t+1}(x_i)=w_t(x_i)\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)^{1-I(\hat{y}^t(x_i)\neq y(x_i))}. \tag{13}$$

对于错分的非单调一致样本,减小其权重:因为非单调一致的样本会影响分类器的精确度,故令其权重直接为 0.

更新权重后的训练集用 U' 表示,在 U' 上采样新的训练子集 U_t. U_t 中的样本由 2 部分组成:一部分 αM 个样本($\alpha\in(0,1)$)从 U_{t-1} 上随机抽取;另一部分 $(1-\alpha)M$ 个样本从 U' 上未被抽取的数据集 $U-\bigcup_{k=1}^{t-1}U_k$ 中随机抽取,每次抽取时保证 U_t 中每个类别的样本数与 M 的比例为 p_l,$l=1,2,\cdots,s$. 迭代上述过程,直到训练集上未被选到的样本数量小于 M 时停止.

由于每次迭代产生 1 个训练子集,每个训练子集对应生成 1 棵决策树,因此迭代执行的次数就是决策森林中决策树的数目. 当 M 确定后,每次从未抽取过的数据集中抽取的样本个数 $(1-\alpha)M$ 也随之确定,进而可以确定总迭代次数. 也就是说,决策森林中决策树的数目可以通过确定每个训练子集的样本规模自动确定.

训练子集的采样算法总结如下.

算法 2. 训练子集的采样算法.

输入:单调数据集 U、参数 Z,E,θ,e,α;

输出:h 个训练子集.

Step1. 用算法 1 确定训练子集的样本规模 M.

Step2. 对于单调数据集 U,按式(9)计算其中各个类别的比例 p_l;

Step3. 按式(10)为训练集中每个样本的权重赋初值;

Step4. 在 U 上随机选择训练子集 U_1,并保证其中各类别样本数占该训练子集中所有样本数的相应比例为 p_l;

Step5. 按以下步骤,迭代生成决策树,直到训练集 U 上未被抽取的样本数小于 M 时结束循环,并记录迭代执行的次数 h;

Step5.1. 在上一训练子集上用 REMT[9] 方法生成决策树,并按式(11)计算决策树的加权误差;

Step5.2. 对错分样本按式(12)计算其单调一致性,对给定参数 e,若 $D(x_i)\geqslant e$,则按式(13)更新 x_i 的权重,否则权重为 0;

Step5.3. 在 U_{t-1} 上抽取 αM 个样本,在 U 上未被抽取的样本中抽取 $(1-\alpha)M$ 个样本,得到新的训练子集 U_t,并保证其各类别的相应比例为 p_l;

Step5.4. 对 U_t 中的样本权重归一化;

Step6. 返回 h 个训练子集 $U_t(t=1,2,\cdots,h)$.

在每个训练子集 U_t 上,采用适用于单调分类的 REMT 算法[9] 可以生成单棵决策树 T_t,$t=1,2,\cdots,h$.

2.2 决策森林的生成算法

本文提出一种基于决策森林的单调分类算法(monotonic classification method based on decision forest, MCDF),通过计算每棵决策树的规则支持度,获得决策森林最终的分类结果.

对于每棵决策树,设叶子节点 $leaf_i$ 中包含的样本集为 U_{leaf_i},这些样本来自不同的类 $c_1,c_2,\cdots$,定义分类规则在类别 c_k 上的规则支持度为

$$Support(c_k)=\frac{|U_{leaf_{ik}}|}{|U_{leaf_i}|}, \tag{14}$$

其中,$U_{leaf_{ik}}$ 表示叶子节点 $leaf_i$ 中的第 k 类样本集.

规则支持度 $Support(c_k)$ 记录在每棵决策树的每个叶子节点上,集成决策树时,根据每棵树的分类规则和规则支持度确定分类结果:当一个新的样本输入时,让每棵决策树根据自己的分类规则对其投票,即投票给它认为正确的类别,最后选择得票数最多的类作为该样本的最终分类结果.若有 r 个类别的得票数相等($r\geqslant 2$),即决策冲突时,计算每棵决策树中用于判断该样本类别的叶子节点的规则支持度,将这 h 棵决策树在每种类别上的规则支持度加和,取规则支持度之和最大的类别作为该样本的最终分类结果,即该样本的分类结果为

$$c_k=\arg\max_k\Big(\sum_{t=1}^{h}Support(c_k^t)\Big),\quad k=1,2,\cdots,r, \tag{15}$$

其中,$Support(c_k^t)$ 为类别 c_k 在决策树 T_t 上的规则支持度,h 为决策树数目.

本文提出的 MCDF 算法的主要步骤如下.

算法 3. MCDF 算法.

输入:训练集 U_{Train}、测试集 U_{Test}、参数 Z,E,θ,e,α;

输出:测试集 U_{Test} 中所有样本的分类结果.

Step1. 用算法 2 得到训练集 U_{Train} 的 h 个训练子集 $U_t,t=1,2,\cdots,h$;

Step2. 在每个训练子集上用决策树算法 REMT[9] 生成决策树 $T_t,t=1,2,\cdots,h$,按式(14)计算每个叶子节点中各个类别上的规则支持度;

Step3. 对测试集中的一个样本 x,让 h 棵决策树对其分类并投票,得到多个分类结果 $c_k,k=1,2,\cdots$;

Step4. 比较每个类别 c_k 的得票数,按以下步骤决定最终分类结果:

Step4.1. 若具有最多票数的 c_k 唯一,记为 $c_{\max}$,则 $c_{\max}$ 就是该样本的最终类别;

Step4.2. 若同时具有最多票数的类别 c_k 有 r 个($r\geqslant 2$),则按式(15)计算这 r 个类中规则支持度之和最大的类别 c_k,作为该样本的最终类别;

Step5. 对测试集 U_{Test} 中的所有样本分类后返回分类结果.

2.3　时间复杂度分析

由于决策树数目可以提前计算,因此,MCDF 方法构建决策森林的时间主要由采样时间和生成决策树的时间 2 部分构成.

采样过程中,样本在特征集与类标签上的包含度可以在采样前提前计算,不需占用采样时间.而计算决策树的加权误差、更新样本权重和抽样过程都需要分别遍历 1 次 M 个样本,因此单棵决策树的采样过程的时间复杂度为 $O(M)$.

生成决策树时,需要分别考虑非叶子节点和叶子节点上的时间复杂度.

1) 在第 i 个非叶子节点上,要依次考虑 m 个特征,以特征值域中的 v 个取值作为分裂点,求每个分裂点下的 RMI,该时间复杂度为 $O(mv)$.计算 RMI 时,对训练集中的每个样本,都需要遍历所有样本比较优劣关系,即该过程的时间复杂度为 $O(n_i^2)$.因此,非叶子节点 i 上求分裂规则的时间复杂度为 $O(mvn_i^2)\leqslant O(mvM^2)$.

2) 在第 j 个叶子节点上,需要遍历该节点中的所有样本求其所包含的每个类别的支持度,故叶子节点上的时间复杂度为 $O(n_j)\leqslant O(M)$.

假设决策森林中的非叶子节点和叶子节点数分别为 k_1 和 k_2,则生成决策树的时间复杂度为 $O(k_1mvM^2+k_2M)$.

综上,设 MCDF 方法构建的决策森林中决策树数目为 h 个,那么 MCDF 方法的时间复杂度为 $O(hM+k_1mvM^2+k_2M)$.

而 REMT 算法使用原训练集中的所有样本训练决策树,没有采样过程,只考虑生成决策树的时间复杂度:与 MCDF 方法中生成决策树的时间复杂度分析同理,一个非叶子节点上的时间复杂度为 $O(mvn^2)$(n 表示原训练集的样本数量),叶子节点上的时间复杂度为 $O(n)$,设决策树中的非叶子节点和叶子节点数分别为 k_1' 和 k_2',则 REMT 算法的时间复杂度为 $O(k_1'mvn^2+k_2'n)$.

由以上分析可知,本方法中虽然增加了采样过程,但采样的时间复杂度仅为 $O(hM)$,而通过采样可以将时间复杂度中的样本数量 M 由原训练集的样本数量 n 减少为采样后的 M,而 $M\ll n$.

3　实验结果及分析

3.1　实验数据集及评价指标

本文分别在 4 个人造数据集、7 个 UCI 及 1 个真实数据集上进行实验.4 个单调的人造数据集构造[8] 为

$$f(x_1,x_2)=x_1+\frac{1}{2}(x_2^2-x_1^2), \tag{16}$$

其中,$x_1,x_2\in[0,1]$.

每个数据集都有 1 000 个样本、2 个特征,类别数依次为 2 类、4 类、6 类和 8 类,其散点图如图 1 所示:

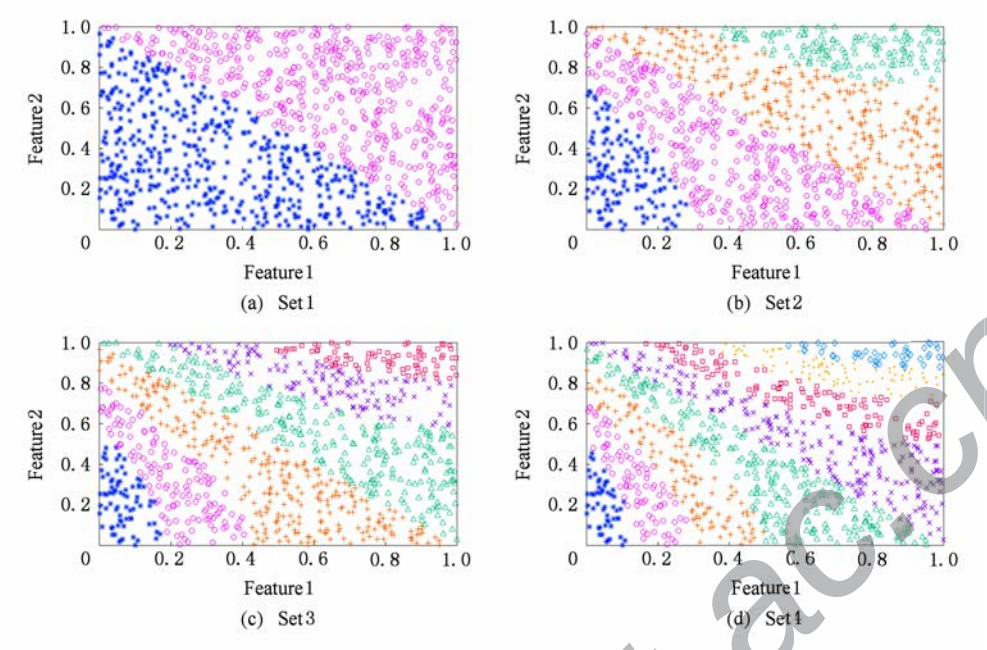

Fig. 1 The artificial datasets

图 1 人造数据集

本文还使用表 1 所示的 8 个数据集验证算法，除 Score 为山西大学学生成绩的真实数据集外，剩下的 7 个为 UCI 数据集. 其中 Wine, Score 为数值型数据集，Breast-w, Car 为符号型数据集，其他 4 个为既包含符号型特征也包含数值型特征的数据集.

Table 1 The Datasets Description

表 1 数据集描述

UCI Dataset	Number of Samples	Number of Features (Number of Numeric Features|Number of Nominal Features)	Number of Classes
Wine	178	13(13|0)	3
Score	512	25(25|0)	4
Breast-w	683	9(0|9)	2
Car	1 728	6(0|6)	4
German	1 000	20(7|13)	2
Adult	500	14(6|8)	2
Australian	690	14(6|8)	2
CPU	209	7(2|5)	5

在每个数据集上进行十折交叉验证. 使用分类准确率和平均绝对误差作为预测能力的评价标准.

分类准确率 CA 定义为

$$CA=\frac{\sum_{x_i\in U} I(\hat{y}(x_i),y(x_i))}{|U|}, \tag{17}$$

其中，$I(\hat{y}(x_i),y(x_i))=\begin{cases}1, & \hat{y}(x_i)=y(x_i);\\ 0, & \hat{y}(x_i)\neq y(x_i).\end{cases}$

平均绝对误差 MAE 定义为

$$MAE=\frac{\sum_{x_i\in U}|\hat{y}(x_i)-y(x_i)|}{|U|}, \tag{18}$$

其中，$y(x_i)$为样本 x_i 的实际类，$\hat{y}(x_i)$为样本 x_i 在分类器上的预测类.

此外，用决策树的平均深度 $Depth$ 作为分类规则长度的评价标准. 实验设备是 1 台配置为3.60 GHz-4 核 CPU、8 GB 内存的计算机，实验平台是 MyEclipse 2015CI. 相关参数设置如下：$Z=1.96$, $E=1.2$, $\theta=0.05$, $e=0.1$, $\alpha=0.5$.

3.2　MCDF 与 REMT 算法的比较

Hu 等人提出的 REMT 单调分类算法[9]是单调分类问题中较为经典的算法,因此本文首先比较 MCDF 与 REMT 单调分类算法的分类精度、平均绝对误差和决策树深度.

图 2 和图 3 分别为 2 种算法在人造数据集和 UCI 及真实数据集上的实验结果.

从图 2 和图 3 可以看出,除人造数据集 Set3 以外,MCDF 方法在所有数据集上都比 REMT 算法的分类准确率高,平均绝对误差低;由于训练子集的规模比原训练集的规模小,所以在包括 Set3 在内的所有数据集上,MCDF 方法都比 REMT 算法的决策树深度小.

在人造数据集 Set3 上,MCDF 方法在 *CA* 和平均绝对误差 *MAE* 上比 REMT 算法低约 0.4%. MCDF 方法在生成多个决策树的迭代过程中,除了每次采样不同的训练样本外,还通过判断错分样本的单调一致程度来改变每棵决策树中训练样本的权重,以增加各个决策树之间的差异度,从而提高决策森林的分类精度. 而由于构造的人造数据集具有较强的单调一致性,一方面单调一致的数据集中不会有权重为 0 的样本出现,另一方面,单棵决策树在单调一致的数据集上每次迭代得到的错分样本数量较少,即权重改变的样本数较少,造成了各个决策树之间的样本权重差异度不大,所以 MCDF 方法的优越性没有得到充分体现;而 REMT 算法由于使用了全部的训练对象,在人造数据集上具有较好的分类性能. 实际上,在人造数据集上,2 种方法的分类性能差别不大:在 Set1,Set2,Set4 上 MCDF 方法精度比 REMT 算法提高不多(分别为 0.9%, 0.6% 和 1.8%),在 Set3 上低于 REMT 算法的差距也很小(0.4%).

根据实验结果,在人造数据集和 UCI 及真实数据集上,MCDF 方法比 REMT 算法的分类准确率分别平均提高了 0.72% 和 4.14%,平均绝对误差分别平均降低了 0.73% 和 4.21%,而且由于使用了较小规模的训练子集,MCDF 方法得到的决策树深度在人造数据集和 UCI 及真实数据集上分别平均减少了 1.6 和 3.21,因此本文提出的方法可以提高单调分类的性能,缩短分类规则的长度.

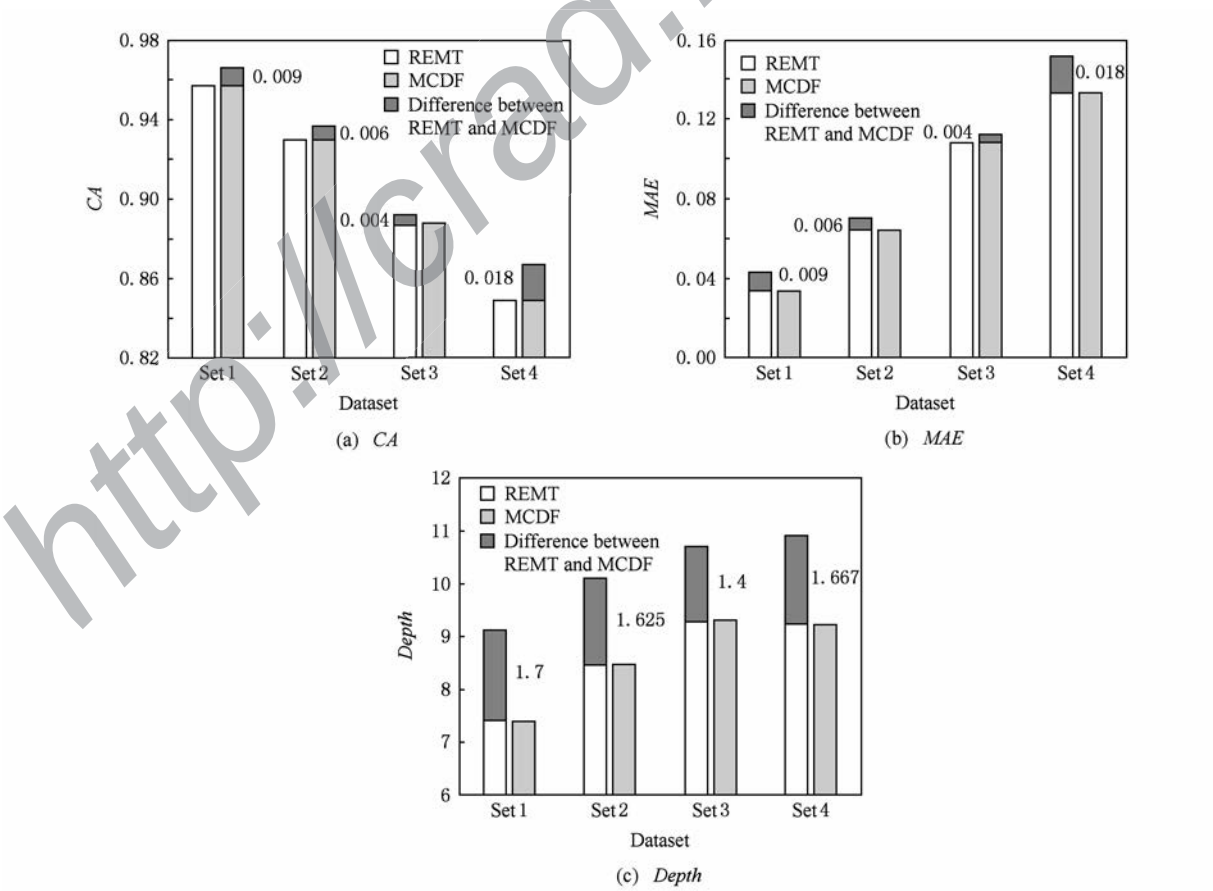

Fig. 2　The experimental result comparison of REMT and MCDF on the artificial datasets

图 2　人造数据集上 REMT 算法与 MCDF 方法的实验结果比较

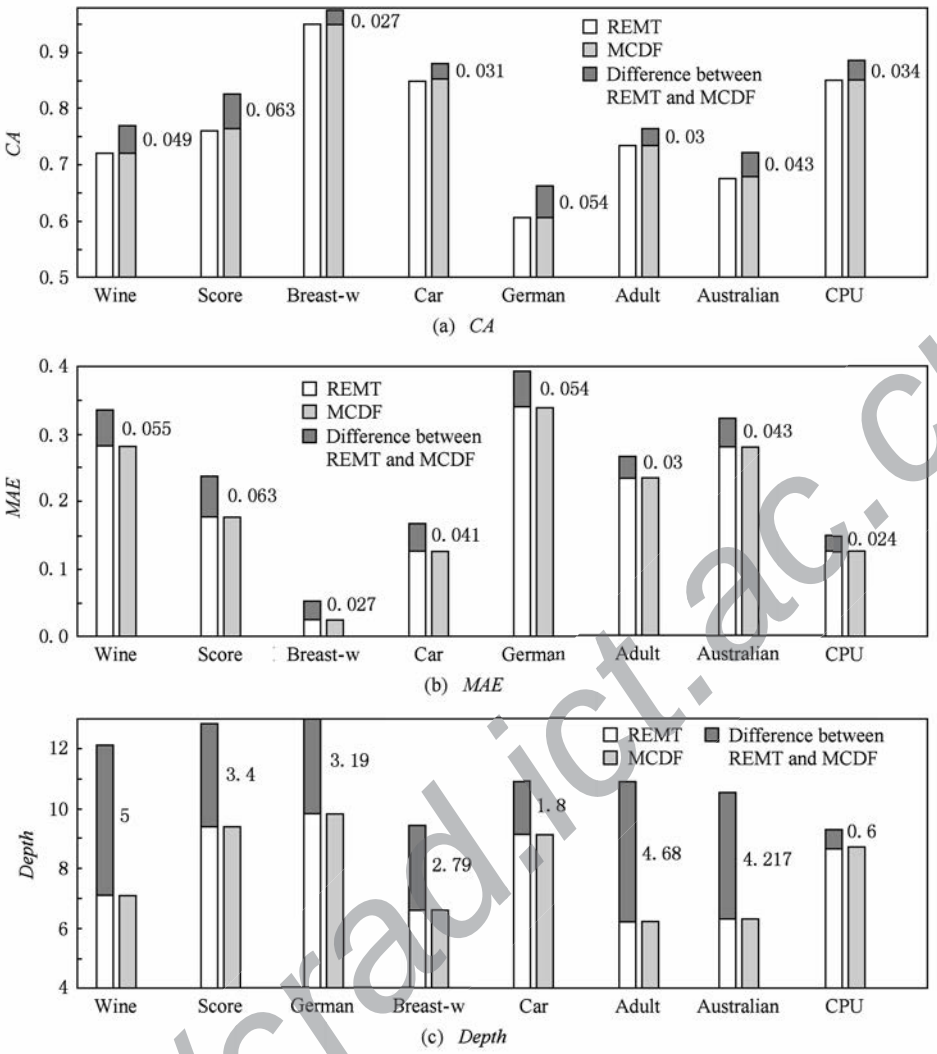

Fig. 3 The experimental result comparison of REMT and MCDF on the UCI and real datasets

图 3 UCI 及真实数据集上 REMT 算法与 MCDF 方法的实验结果比较

3.3 MCDF 与 Adaboost.M1 算法的比较

MCDF 与经典的集成学习方法 Adaboost.M1 算法[14]进行了比较,表 2 是 2 种方法在 12 个数据集上的准确率和平均绝对误差比较.

从表 2 可以看出,除 Australian 以外的其他 11 个数据集上,MCDF 比 Adaboost.M1 的分类准确率高、平均绝对误差低.因为 Adaboost.M1 算法是根据基分类器的错误率改变错分的训练样本的权重,进而构造不同的训练子集,生成决策树,而对于单调分类问题,由于训练集中存在非单调一致的样本,在 Adaboost.M1 算法的迭代过程中,这些样本的权重逐渐增大,降低了基分类器的准确率,使得 Adaboost.M1 算法的集成结果较差.而本文算法中根据 $D(x_i)$判断样本 x_i 的单调一致性,避免采样非单调一致的样本,同时提高采样得到的训练子集的差异性,所以取得了较好的集成效果.

在数据集 Australian 上,MCDF 方法在 *CA* 和平均绝对误差 *MAE* 上都比 Adaboost.M1 算法低 2.3%,其原因首先与数据集 Australian 本身的特点有关.根据文献[12]中有序数据集上特征对类别的依赖度的定义可以求得:数据集 Australian 上的特征依赖度为 0.998,接近于 1,也就是说几乎所有样本都可以被一致地分配类别,这与人造数据集上的情况相似.其次,MCDF 方法在数据集 Australian 上的迭代过程中较少出现权重改变较大的样本,降低了 MCDF 方法创建的决策森林中各个决策树之间的差异度,使得 MCDF 方法的性能受到影响;而数据集 Australian 中噪声较少的特点恰好避开了 Adaboost.M1 算法对噪声比较敏感的弱点,使其分类效果不受影响.另外,数据集 Australian 的类别数为 2,也使得 MCDF 方法针对多类别之间的有序关系上的优势难以得到体现,而 Adaboost.M1 每次都

使用了全部训练样本生成决策树,具有相对较好的分类性能.以上3个因素共同造成了MCDF在Australian上的分类性能略差于Adaboost.M1算法.但是与Set3一样,虽然MCDF方法的分类精度比Adaboost.M1略有下降,但其运行效率比Adaboost.M1算法高.

Table 2　The *CA* and *MAE* Result Comparison of Adaboost.M1 and MCDF on the Datasets

表2　Adaboost.M1算法与MCDF方法的准确率和误差比较

Dataset	*CA*		*MAE*	
	Adaboost.M1	MCDF	Adaboost.M1	MCDF
Set1	0.956±0.024	**0.966±0.022**	0.044±0.024	**0.034±0.022**
Set2	0.929±0.020	**0.936±0.026**	0.071±0.020	**0.064±0.026**
Set3	**0.888±0.030**	**0.888±0.038**	**0.112±0.030**	**0.112±0.038**
Set4	0.853±0.019	**0.867±0.024**	0.147±0.019	**0.133±0.024**
Wine	0.739±0.104	**0.770±0.125**	0.330±0.147	**0.281±0.129**
Score	0.797±0.063	**0.824±0.046**	0.203±0.063	**0.176±0.046**
Breast-w	0.959±0.014	**0.975±0.017**	0.041±0.014	**0.025±0.017**
Car	0.843±0.025	**0.880±0.031**	0.175±0.032	**0.126±0.030**
German	0.612±0.047	**0.661±0.053**	0.388±0.047	**0.339±0.053**
Adult	0.748±0.047	**0.764±0.049**	0.252±0.047	**0.236±0.049**
Australian	**0.743±0.037**	0.720±0.048	**0.257±0.037**	0.280±0.048
CPU	0.861±0.057	**0.885±0.070**	0.139±0.063	**0.125±0.070**

3.4　REMT,Adaboost.M1与MCDF的运行时间

REMT,Adaboost.M1与MCDF方法在运行时间上进行了比较,表3是3种方法在12个数据集上的实验结果.

Table 3　The Running Time Comparison of REMT, Adaboost.M1 and MCDF

表3　REMT,Adaboost.M1与MCDF方法的运行时间比较　　s

Dataset	REMT	Adaboost.M1	MCDF
	Running Time	Running Time	Running Time (Sampling Time)
Set1	3.939	54.158	**0.416**(0.005)
Set2	4.146	58.844	**0.448**(0.005)
Set3	4.162	57.830	**0.450**(0.005)
Set4	4.413	58.079	**0.453**(0.008)
Wine	0.138	1.404	**0.030**(0.002)
Score	0.212	2.950	**0.098**(0.002)
Breast-w	0.136	1.753	**0.068**(0.002)
Car	0.161	2.102	**0.095**(0.006)
German	0.283	3.245	**0.059**(0.003)
Adult	0.069	0.876	**0.025**(0.002)
Australian	0.119	1.230	**0.061**(0.006)
CPU	0.053	0.441	**0.031**(0.002)

从表3可以看出,运行效率上,MCDF方法的执行时间比REMT算法、Adaboost.M1算法都有很大提高,尤其在数值型数据集上,如Set1～Set4,效果非常显著.按照2.3节中的分析可知,生成每棵决策树的时间复杂度为$O(k_1mvM^2+k_2M)$,由于Set1～Set4为数值型数据集,所以其特征的值域取值有n_i种,即$v=n_i$,以根节点为例,数值型数据集根节点上的时间复杂度就为$O(mM^3)$,那么与REMT和Adaboost.M1算法每次都使用训练集中的所有对象生成单棵决策树相比,如果MCDF方法的样本量减少为原来的20%,根节点上的运行时间就可以减少为原来的1/500,同理,其他节点上的运行时间也都有不同程度的降低,而采样过程所增加的时间花销却很小.因此,尽管REMT算法只训练1棵决策树,但其时间复杂度为$O(k'_1mvn^2+k'_2n)$,而$M\ll n$,所以其运行速度仍然比MCDF方法慢,而Adaboost.M1算法所需时间为REMT的h_A倍(h_A为Adaboost.M1算法构造的基分类器数量),运行速度更慢.由实验结果可知,在数值型的人造数据集上,MCDF方法的平均时间花销分别约为REMT和Adaboost.M1算法的1/9和1/130;在UCI及真实数据集上,MCDF方法的平均时间花销提高最多的是German数据集,分别约为REMT和Adaboost.

M1 的 1/5 和 1/55,即使在提高最少的 CPU 数据集上,MCDF 方法的时间花销也仅约为 REMT 和 Adaboost.M1 的 1/2 和 1/14.因此,本文提出的方法有效地提高了运行效率.

3.5 MCDF 在较大规模数据集上的有效性

由于 UCI 中单调数据集的规模较小,为了验证 MCDF 在较大规模数据集上的有效性,用以下函数构造单调数据集:

$$f(x_1,x_2,\cdots,x_m)=\frac{2}{m}\left[\frac{1}{10}x_1+\frac{1}{2}\left(\left(\frac{1}{10}x_2\right)^2-\left(\frac{1}{10}x_1\right)^2\right)+\frac{1}{10}x_3+\frac{1}{2}\left(\left(\frac{1}{10}x_4\right)^2-\left(\frac{1}{10}x_3\right)^2\right)+\cdots+\frac{1}{10}x_{m-1}+\frac{1}{2}\left(\left(\frac{1}{10}x_m\right)^2-\left(\frac{1}{10}x_{m-1}\right)^2\right)\right], \quad (19)$$

其中,$x_1,x_2,\cdots,x_m$ 是 m 个服从均匀分布的独立随机变量(m 为偶数),表示 m 个特征,其中 $x_1,x_2,\cdots,x_{\frac{m}{2}}\in[0,10]$,$x_{\frac{m}{2}+1},x_{\frac{m}{2}+2},\cdots,x_m\in\{0,1,2,\cdots,10\}$,将该函数的值域[0,1]分为 s 个区间:$\left[0,\frac{1}{s}\right]$,$\left(\frac{1}{s},\frac{2}{s}\right],\cdots,\left(\frac{s-1}{s},1\right]$,分别对应 s 个类别$\{1,2,\cdots,s\}$.数据集的详细信息如表 4 所示:

Table 4 The Large Dataset Description

表 4 大规模数据集描述

Large Dataset	Number of Samples	Number of Features (Number of Numeric Features|Number of Nominal Features)	Number of Classes
LargeSet1	100 000	8(4|4)	5
LargeSet2	100 000	12(6|6)	8
LargeSet3	100 000	48(24|24)	8
LargeSet4	500 000	8(4|4)	5
LargeSet5	500 000	12(6|6)	8
LargeSet6	500 000	48(24|24)	8

在该数据集上,MCDF 方法的实验结果如表 5 所示.而 REMT 和 Adaboost.M1 算法在数据集 LargeSet1,LargeSet2 上用 MCDF 方法的 10 倍运行时间仍无法得到结果;在 LargeSet3~LargeSet6 这 4 个数据集上,也无法在可接受的时间内得到结果.

Table 5 The Experimental Result on the Large Dataset

表 5 大规模数据集实验结果

Large DataSet	*CA*	*MAE*	*Depth*	Sampling Time/s	Total Running Time/s
LargeSet1	0.854	0.148	12.9	1.060	696.469
LargeSet2	0.767	0.233	13.8	1.524	2 271.657
LargeSet3	0.804	0.196	14.5	5.829	9 766.699
LargeSet4	0.877	0.123	14.8	20.865	35 269.533
LargeSet5	0.782	0.218	15.2	21.041	62 898.782
LargeSet6	0.823	0.177	15.9	46.784	184 350.219

MCDF 方法在大规模数据集上的有效性在理论上的分析如下:根据 2.3 节的时间复杂度分析可知,MCDF 方法生成决策树的时间复杂度为 $O(k_1mvM^2+k_2M)$,而 REMT 生成决策树的时间复杂度为 $O(k'_1mvn^2+k'_2n)$.因为数据集中包括符号型和数值型特征,对于符号型特征,式中的 v 为一个常数,而对于数值型特征,式中的 v 就是每个节点中样本的数量.例如在根节点上,MCDF 方法在数值型特征上的时间复杂度为 $O(M^3)$,而 REMT 算法为 $O(n^3)$,以本节实验中样本总量为 10 万的数据集 LargeSet1 为例,由 2.1 节中确定训练子集规模 M 的方法可以算出该数据集的 M 为 900,进而得出决策树数目为 220 棵.由于这里的 M 小于原数据集数据量的 $\frac{1}{100}$,所以在数值型特征上,MCDF 方法所需的时间仅为 REMT 算法的 $\left(\frac{1}{100}\right)^3\times220$,在符号型特征上,MCDF 方法所需的时间也仅为 REMT 算法的 $\left(\frac{1}{100}\right)^2\times220$.这样,MCDF 方法所需总时间最多为 REMT 算法的 $\frac{1}{45}$,那么当 MCDF 方法在该数据集上的运行时间为 696.469 s 时,REMT 算法的运行时间应大于 8.7 h.对于特征数量或样本数量更大的数据集(LargeSet2~LargeSet6),MCDF 方法和 REMT 算法的运行时间差距将会非常大.而 Adaboost.M1 算法则需要比 REMT 算法更长的时间,因为它每次都使用了全部训练样本生成决策树,所以它所需的时间为 REMT 的 h_A 倍.由以上分析

可知,MCDF 方法在处理较大规模的数据集上是非常有效的.

4　结束语

本文提出的 MCDF 方法针对单调的数据集,用基于数据集的构造方式构建决策森林,解决单调分类问题.该方法既可处理符号型数据,又可处理数值型数据以及包含符号型和数值型的混合数据;设计的采样策略,在保持较高分类精度的同时大大提高了运行效率;提出的规则支持度有效地保证了单调分类的性能;此外,该方法还可以解决大规模数据集上的单调分类问题.本文通过数据集各个特征上的取值来自动确定决策树的数目,可以避免人工干预,但需要遍历所有样本来获得对决策树的评价,对算法效率有一定影响.能否根据决策树本身的结构选择部分决策树来构成决策森林还需要进一步研究.

参　考　文　献

[1] Greco S, Matarazzo B, Slowinski R. Rough approximation by dominance relations [J]. International Journal of Intelligent Systems, 2002, 17(2): 153-171

[2] Sai Ying, Yao Yiyu, Zhong Ning. Data analysis and mining in ordered information tables [C] //Proc of IEEE ICDM'01. Los Alamitos, CA: IEEE Computer Society, 2001: 497-504

[3] Kotlowski W, Dembczynski K, Greco S, et al. Stochastic dominance based rough set model for ordinal classification [J]. Information Sciences, 2008, 178(21): 4019-4037

[4] Hu Qinghua, Yu Daren, Guo Maozu. Fuzzy preference based rough sets [J]. Information Sciences, 2010, 180(10): 2003-2022

[5] Qian Yuhua, Dang Chuangyin, Liang Jiye, et al. Set-valued ordered information systems [J]. Information Sciences, 2009, 179(16): 2809-2832

[6] Ben-David A. Automatic generation of symbolic multiattribute ordinal knowledge-based DSSs: Methodology and applications [J]. Decision Sciences, 1992, 23(6): 1357-1372

[7] Hu Qinghua, Guo Maozu, Yu Daren, et al. Information entropy for ordinal classification [J]. Science China: Information Sciences, 2010, 53(6): 1188-1200

[8] Hu Qinghua, Pan Weiwei, Zhang Lei, et al. Feature selection for monotonic classification [J]. IEEE Trans on Fuzzy Systems, 2012, 20(1): 69-81

[9] Hu Qinghua, Che Xunjian, Zhang Lei, et al. Rank entropy based decision trees for monotonic classification [J]. IEEE Trans on Knowledge and Data Engineering, 2012, 24(11): 2052-2064

[10] Che Xunjian. Ordinal decision tree based fault level detection [D]. Harbin: Harbin Institute of Technology, 2011 (in Chinese)
(车勋建. 基于有序决策树的故障程度诊断研究[D]. 哈尔滨:哈尔滨工业大学, 2011)

[11] Breiman L. Random forest [J]. Machine Learning, 2001, 45(1): 5-32

[12] Hu Qinghua, Yu Daren. Applied Rough Computing [M]. Beijing: Science Press, 2012 (in Chinese)
(胡清华, 于达仁. 应用粗糙计算[M]. 北京:科学出版社, 2012)

[13] Breiman L. Bagging predictors [J]. Machine Learning, 1996, 24(2): 123-140

[14] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting [J]. Journal of Computer and System Sciences, 1997, 55(1): 119-139

[15] Wang Wenjian, Men Changqian. Support Vector Machine Modeling and Application [M]. Beijing: Science Press, 2014 (in Chinese)
(王文剑, 门昌骞. 支持向量机建模及应用[M]. 北京:科学出版社, 2014)

[16] Liang Jiye, Wang Feng, Dang Chuangyin, et al. An efficient rough feature selection algorithm with a multi-granulation view [J]. International Journal of Approximate Reasoning, 2012, 53(6): 912-926

Xu Hang, born in 1987. PhD candidate. Her main research interests include machine learning and service computing.

Wang Wenjian, born in 1968. PhD. Professor, PhD supervisor. Senior member of CCF. Her main research interests include neural networks, support vector machine, machine learning and intelligent computing.

Ren Lifang, born in 1976. PhD candidate. Lecturer. Her main research interests include machine learning and service computing.