

DOI: 10.11830/ISSN.1000-5013.201710002



# 采用口袋算法构造的多类别 决策树模型

王子玥, 谢维波, 李斌

(华侨大学 计算机科学与技术学院, 福建 厦门 361021)

**摘要:** 采用开放应用架构(OAA)准则训练多个二分类感知机,以 Gini 指数筛选最优的方法构建二叉决策树.推算说明感知机多分类准则在每个树节点上对空间划分的局限性,将基于口袋算法的二叉树与多叉树在 8 个 UCI 数据集上进行比较,并与单变量决策树 CART 和 C4.5 的结果进行对照.结果表明:采用口袋算法基于 OAA 方法构建的二叉树,在准确率和空间划分的可解释性上优于基于经典多分类准则构建的多叉树.

**关键词:** 感知机多分类; 开放应用架构; 口袋算法; Gini 指数; 决策树

**中图分类号:** TP 311      **文献标志码:** A      **文章编号:** 1000-5013(2019)01-0121-07

## Model Construction of Multi Class Decision Tree Using Pocket Algorithm

WANG Ziyue, XIE Weibo, LI Bin

(College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China)

**Abstract:** This paper attempts to use open application architecture (OAA) criteria training several binary perceptron and then using Gini index to screen the best one to construct two fork decision tree. Theoretically explain the structural limitations of the perceptron classification criterion on the spatial partitioning at each tree node. Based on the experimental results of 8 UCI datasets we compared pocket algorithm two fork tree and multifork tree. We also compared the experimental results with univariate decision tree CART and C4.5. The results show the tree build by pocket algorithm and OAA method is superior to the others in both the accuracy rate and the interpretability of spatial division.

**Keywords:** perceptron multi classification; open application architecture; pocket algorithm; Gini index; decision tree

口袋算法是基于感知机的更新公式、适用于线性不可分的二分类数据的算法,在训练过程中,统计最长的连续正确运行次数,每当找到一个新的最大次数,就验证当前的感知机向量在数据集上的准确率,口袋算法会保存准确率最高的感知机向量,并作为结果输出.已有的成果表明,口袋算法在有限的迭代次数内,能够以概率 1 找到最优解<sup>[1]</sup>.在决策树的构建过程中,经典的 ID3, C4.5, CART 对于样本空间的划分都是轴平行的.早期研究采用基于各种线性模型构建非轴平行的多变量决策树的方法,如基于 Fisher 判别准则构建的多变量决策树<sup>[2]</sup>,基于贪心思想采取随机扰动的 OC1 算法<sup>[3]</sup>,以及其他方式构建的多变量决策树<sup>[4-6]</sup>.采用口袋算法构建决策树的过程中,主要存在如何用多类别的数据在单个节点

**收稿日期:** 2017-10-10

**通信作者:** 谢维波(1964-),男,教授,博士,主要从事信号处理、视频图像分析的研究. E-mail: xwblxf@hqu.edu.cn.

**基金项目:** 国家自然科学基金资助项目(61271383); 华侨大学研究生科研创新能力培育计划资助项目(16113 14016)

上训练二分类感知机的问题. 基于感知机学习方式的多变量决策树中,通常是利用经典的感知机多分类准则构建多叉树<sup>[7]</sup>,而支持向量机(SVM)一类直接求解的线性模型则需要采用组合的方法建立二叉决策树<sup>[8-9]</sup>. 如同在单变量决策树中的多叉树和二叉树的区别一样,多叉树存在决策速度过快、树深度较低的特点. 在感知机的多分类准则中,如果训练的类别数目超过了两类,那么,训练出的多类别感知机虽然可以将任意一个样本无歧义地划分到某一类,但是其决策过程采用的是选择计算出的函数值最大的一类作为决策结果,意味着函数本身不再是样本空间划分的边界表示. 本文在单个树节点上采用开放应用架构(open application architecture,OAA)的思想,每次对  $c$  个类别单独训练;然后,采用 Gini 指数筛选最优结果,迭代此过程生成决策树.

1 口袋算法

设  $w^*$  为“口袋”向量, $\alpha^*$  为  $w^*$  的分类准确率, $s^*$  为  $w^*$  对应的正确运行次数, $w$  为更新过程中的感知机向量, $\alpha$  为  $w$  的分类准确率, $s$  为  $w$  对应的正确运行次数, $n$  为训练集样本数目, $d_i$  表示训练集第  $i$  个样本, $t$  为设定的对数据集的最大遍历轮数. 口袋算法流程有如下 6 个步骤.

- 步骤 1 初始“口袋”向量为空,初始化  $s=0$ ,设定初始  $w$ .
- 步骤 2 对训练集随机重排.
- 步骤 3 读取第  $i$  个样本, $i=1,2,\cdots,n$ .
- 步骤 4 若  $w$  正确划分  $d_i$ , $s=s+1$ ;若  $w$  错误划分  $d_i$  且  $s<s^*$ ,则按照感知机的更新规则更新  $w$ ;若  $w$  错误划分  $d_i$  且  $s>s^*$ ,验证当前  $w$  对训练集的分类准确率  $\alpha$ ,并对比  $\alpha^*$  和  $\alpha$ . 如果  $\alpha<\alpha^*$ , $s$  设置为 0,按照感知机的更新规则更新  $w$ ;如果  $\alpha>\alpha^*$ ,将  $s$  保存为  $s^*$ , $\alpha$  保存为  $\alpha^*$ , $w$  保存为  $w^*$ , $s$  设置为 0. 然后,按照感知机的更新规则更新  $w$ .
- 步骤 5 如果数据集未全部训练完毕,回到步骤 3,开始下一个样本的训练.
- 步骤 6 如果数据集全部样本训练完毕,且全样本均正确划分或达到了预设的  $t$  轮训练,则算法止;否则,回到步骤 2,开始下一轮训练.

2 感知机多分类准则

2.1 多分类学习过程

基于感知机的多类线性判别函数是在训练过程中对  $c$  个类别设定  $c$  个判别函数. 感知机多类别学习过程如下:1) 初始化向量  $w_j, j=1,2,\cdots,c$ ,设置  $m=0$ ;2) 考察某个样本  $x_i\in c_j$ ,若对任意的  $p(p\neq j)$  均存在  $x_i^T w_j(m)>x_i^T w_p(m)$ ,说明分类正确,所有权向量不变;若存在某个类  $p$ ,使  $x_i^T w_j(m)\leq x_i^T w_p(m)$ ,则选择使得  $x_i^T w_p(m)$  最大的类别  $p$  按照下式进行权向量更新,即

$$\left. \begin{aligned} w_j(m+1) &= w_j(m) + \eta x_i, \\ w_p(m+1) &= w_p(m) - \eta x_i, \\ w_l(m+1) &= w_l(m), \quad l \neq j, p. \end{aligned} \right\} \tag{1}$$

将式(1)应用于口袋算法建立多叉树时,用以替换前述口袋算法中的感知机更新规则,从而实现利用口袋算法建立多叉树的目标.

2.2 空间划分别

实际训练出的多分类感知机并不表示真实的边界,因此,构建一个简单的三类别数据集用以说明情况. 在二维平面上构造 3 个类别的样本:第 1 类 35 个样本随机分布在  $x\in[0,1],y\in[0,1]$  区域内;第 2 类 35 个样本随机分布在  $x\in[1.2,2.2],y\in[0,1]$  内;第 3 类 30 个样本随机分布在  $x\in[2.4,3.4],y\in[0,1]$  区域内. 可以看出,这是由 100 个样本组成、完全线性可分的数据集,在运用口袋算法使用感知机多分类准则进行训练时,训练结果如图 1 所示. 图 1 中:3 条直线为学习所得的 3 个感知机向量,3 条直线对数据集的划分非常差,这 3 个函数对数据集的分类准确率在 98%~100%之间波动.

采用的二叉树建树策略生成的决策树,如图 2 所示. 其通过 2 个节点生成的 2 条直线将 3 类别数据划分开,且学习所得的 2 条直线就表示了对空间的真实划分. 这也说明采用感知机多分类准则学习到的

多类别函数并不实际表示对空间区域划分的边界。

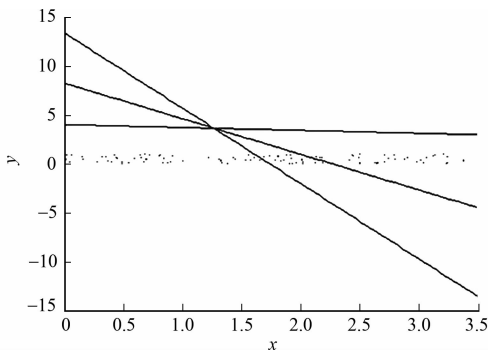


图 1 多分类感知机

Fig. 1 Multi classification perceptron

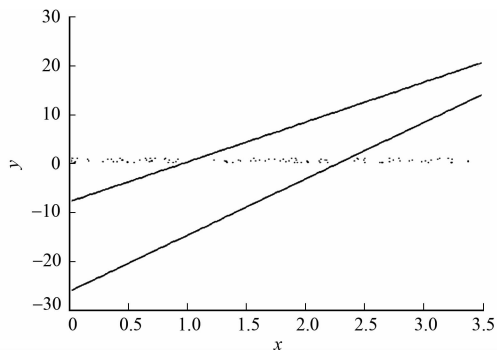


图 2 二分类感知机树

Fig. 2 Binary perceptron tree

对于多分类感知机来说,这种空间边界是可求得的.以三类别的感知机为例,设训练完成的 3 个函数分别为  $w_1, w_2, w_3$ ,若某一样本被划分至第  $j$  类,则意味着对任意  $p(p \neq j)$  均有  $x_i^T w_j(m) > x_i^T w_p(m)$ .由此可以计算出在三分类情况下的分类边界为

$$(w_1 - w_2)x^T = 0, \quad (w_2 - w_3)x^T = 0, \quad (w_3 - w_1)x^T = 0. \tag{2}$$

记式(2)的 3 个函数分别为  $a, b, c$  式.归属第一类时,  $a$  式  $> 0, c$  式  $< 0$ ;第二类要求  $a$  式  $< 0, b$  式  $> 0$ ;第三类要求  $b$  式  $< 0, c$  式  $> 0$ .也就是说,三分类情况是由这样推算出的其中 2 个超平面共同限制的区域构成,这种划分会覆盖全样本空间,当然这样的分类超平面作出的分类边界不是无限延伸的,如图 3 所示.由于  $w$  和  $x$  均为增广向量形式,若式(2)中任意两式表示的是平行向量,则实际为同一分类超平面.因此,这 3 个分类超平面在大于等于二维的样本空间中必然存在可求解.同理,  $c$  个类别的数据在大于等于  $(c-1)$  维的样本空间也会存在解.这样的可求解意味着感知机多分类划分出的多个子空间存在一个或一组共同的边界点,如果在二维平面上表示,就会呈现出图 3 形状的星形划分;而对于用多个二分类取代多分类的边界划分没有此类限制。

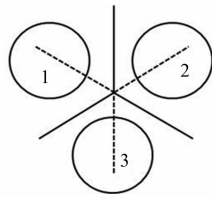


图 3 多分类边界

Fig. 3 Multi class boundary

另一方面,决策树在迭代生成的过程中,其子节点的训练集样本数量和类别都会逐渐减少;而当多分类准则应用在决策树的每个节点上时,若数据集有  $c$  个类,则要求树的每个非叶节点都必须划分出  $c$  个区域;如果子节点的训练集中出现了孤立点和噪声点,那么,它们依然会被划分出对应的区域,这在应用口袋算法生成的决策树中会影响分类性能。

### 3 基于口袋算法的二叉树建树策略

#### 3.1 OAA 准则

若数据集有  $c$  个类,记为  $c = \{c_1, c_2, \dots, c_k\}$ . OAA 策略就是生成  $j$  个分类函数  $\{f_1, f_2, \dots, f_k\}$ ,每一个  $f$  对应一个分类器,而每一个分类器的生成过程都是将某一类  $c_j$  和其余  $(c - c_j)$  类相区分.在生成完所有的  $f$  后,留下最优的一个  $f$  作为被选择的分类函数<sup>[10]</sup>.

#### 3.2 Gini 指数

假设集合  $D$  有  $K$  个类,样本点属于第  $k$  类的概率为  $p_k$ ,则概率分布的基尼指数定义为

$$\text{Gini}(p) = \sum_{k=1}^K p_k(1 - p_k) = 1 - \sum_{k=1}^K p_k^2. \tag{3}$$

Raileanu 等<sup>[11]</sup>证明了采用何种损失函数在决策树的构建过程中影响较小;潘大胜等<sup>[12]</sup>对熵的计算过程加权以提升准确率.为了体现多叉树和二叉树的区别,后续实验统一采用 Gini 指数作为每个节点上选择分类器的损失函数。

#### 3.3 决策树生成步骤

采用口袋算法构建二叉决策树,首先,对数据进行预处理,一是对原始数据的顺序进行随机重排,以减少顺序对结果及口袋算法最长正确运行次数的影响;二是将原始数据进行高斯归一化,记  $f_{i,j}$  为当前

训练集上第  $i$  个样本的第  $j$  个特征的值,则高斯归一化公式为

$$\hat{f}_{i,j} = \frac{f_{i,j} - E(f_j)}{D(f_j)}.$$

(4)

式(4)中: $\hat{f}_{i,j}$ 为对原始数据归一化后的值; $E(f_j)$ 为第  $j$  个特征下所有值的均值; $D(f_j)$ 为第  $j$  个特征下所有值的方差.

对于非数值型数据,需要先将原始数据进行编号,编为  $1 \sim n$  的一组数字;然后,再进行高斯归一化,归一化后可以消除量纲间的差别.

从根节点开始,生成二叉决策树的算法流程有如下 4 个步骤.

**步骤 1** 设当前结点训练集为  $D$ ,若训练集具有  $c$  个类别,则基于 OAA 准则,将输入数据集复制为  $c$  组数据,每组数据具有 2 个类别,分别为第  $c_j$  类和非第  $c_j$  类数据.

**步骤 2** 利用口袋算法在步骤 1 上的  $c$  组数据上分别训练  $c$  个感知机向量,每一个感知机向量的训练在达到设定的最大遍历次数后停止;然后,计算每个感知机分类结果的 Gini 指数,选择 Gini 指数最小的那个感知机作为当前节点的分类器.

**步骤 3** 按照选择出的感知机分类器将  $D$  中数据划分到 2 个子节点中.

**步骤 4** 对两个子节点递归地调用步骤 1,2,3,直至满足预剪枝条件.

在步骤 1 中,2 个子节点上的 Gini 指数仍然按照多类别计算,即  $\text{Gini}(D_1)$  和  $\text{Gini}(D_2)$  中有多个类别的概率,计算式为

$$\text{Gini}(D_i) = \sum_{k=1}^c p_k(1 - p_k) = 1 - \sum_{k=1}^c p_k^2, \quad i = 1, 2,$$

(5)

$$\text{Gini} = \frac{|D_1|}{|D|} \text{Gini}(D_1) + \frac{|D_2|}{|D|} \text{Gini}(D_2).$$

(6)

也就是说,2 个子集上 Gini 的计算仍然是按照数据的原始类别计算.假设当前节点选择第  $c_j$  类生成的感知机作为分类器,将  $D$  分为 2 个子集,那么,在子集上计算 Gini 时,其余  $(c - c_j)$  类的数据依然按照原有类别计算各自的  $p_k$ .这样是为了保证在生成树过程中,当某几类样本都可以被从  $D$  中完全分离时,这些类别对应的分类器所计算出的 Gini 指数不会相同,从而依然能够选择出唯一最优的划分.

使用口袋算法构造二叉树,同时构造多叉树作为对比.为了保证可比性,二者采用相同的数据预处理.在单个结点上的训练过程中,将口袋算法流程中的感知机更新规则替换为感知机多分类准则.生成多叉树的流程如下:1) 设当前结点训练集为  $D$ ,应用感知机多分类准则替换后的口袋算法训练得到  $c$  个感知机向量;2) 将  $D$  中数据送入第一步生成的  $c$  个感知机向量中,以值最大的向量作为该样本对应的类别进行多叉树划分;3) 对多个子节点递归地调用流程 1),2),直至满足预剪枝条件.

4 实验结果及分析

为了验证二叉建树与多叉建树算法的有效性并对比其性能上的区别,采用 8 个 UCI 数据集进行对比实验,增加了经典决策树 CART<sup>[13-15]</sup>和 C4.5<sup>[16]</sup>的实验结果作为经典轴平行决策树进行对比.硬件平台采用 i7-6700HQ CPU,128 G 的硬盘,8 G SSD 内存,实验中不涉及 GPU 的加速运算;软件平台采用 MATLAB 2014a.

采用十折交叉验证,分类准确率取 30 次十折交叉验证结果的平均值,算法花费时间取 30 次十折交叉验证结果的平均值.数据集详细信息,如表 1 所示.表 1 中: $n$  是样本总数; $c$  是样本类别; $k$  是特征数目.表 1 中的数据均为去除含有缺失项的样本之后统计得到,残缺样本不做任何处理,后续实验部分也不将残缺样本计算在内.

4.1 参数设置

所有数据集均采用相同的参数设置.在所提采用口袋算法构造的针对多类别数据的二叉树中,需要

表 1 数据集信息

Tab. 1 Data set information

数据集	$n$	$c$	$k$
Balance	625	3	4
Breast	683	2	9
Glass	214	7	9
Iris	150	3	4
Spambase	4 601	2	57
Segment	2 100	7	19
Sensor	5 456	4	2
Wine	1 599	6	11

人工设置的参数为剪枝策略,每个节点上应用口袋算法时的最大遍历轮数和口袋算法的步长.

剪枝条件用以控制树的生长<sup>[17-18]</sup>,采取决策树中常见的预剪枝策略<sup>[19]</sup>,剪枝策略如下:1) 某一节点上,若占比最大的类别的样本数目比例超过 90%,则此节点不再生成子节点;2) 当某一节点上样本数目低于总体训练集的 5%时,不再生成子节点;3) 生成的子节点中,主要类别若与父节点的主要类别相同且主要类样本占比提升小于 2.5%,则删除此子节点,用父节点输出分类结果.实验中的多叉树、二叉树及作为对比的经典单变量决策树 CART 和 C4.5 均采用此剪枝策略.

口袋算法中,数据集遍历的阈值设置越大,理论上找到的划分就越接近最优解.文中设置的阈值为 40,当训练集遍历轮数达到此值时,算法终止.从实验结果中可以看出,分类准确率方差普遍较小,可以认为算法已达到稳定.

由于感知机向量对样本的划分可以看作是将原本  $n$  维的特征空间映射至一维数轴上,然后按照阈值进行区分.口袋算法的步长设置是在  $c$  个类别中计算各个类别样本的均值中心点,然后计算这些中心点两两之间的距离,选择最小的距离并除以 100 作为学习的固定步长.这样设置是为了在两类之间通过感知机学习过程找出一个合适的阈值位置.记最终步长为 STEP,则步长的计算式为

$$\text{STEP} = \max \left[ \frac{\sum_{i=1}^{n_j} \boldsymbol{d}_{j,i}}{n_j} - \frac{\sum_{i=1}^{n_k} \boldsymbol{d}_{k,i}}{n_k} \right] / 100, \quad j = 1, 2, \cdots, c; \quad k = 1, 2, \cdots, c. \tag{7}$$

式(7)中: $j$  和  $k$  表示类别号; $n_j$  和  $n_k$  表示数据集中第  $j$  类和第  $k$  类的样本数目; $\boldsymbol{d}_{j,i}$  和  $\boldsymbol{d}_{k,i}$  分别表示对应类别下的样本,且这些样本均为经过数据预处理后的样本向量.

4.2 实验分析

对比二叉树、多叉树及两种经典决策树算法在 8 个数据集上的分类性能,结果如表 2 所示.表 2 中:二叉树指的是基于口袋算法和 Gini 指数选择的二叉树模型;多叉树为基于口袋算法和经典感知机多分类准则的多叉树模型;\* \* 表示 0.99 置信度显著性检验. Breast 和 Spambase 是二分类数据集,所以在这两个数据集上构建的二叉树和多叉树是完全相同的.二者的唯一差别在于二叉树算法对各类别的感知机向量是独立训练而后选取最优,而多叉树在这两个训练集上训练出的是正负号相反而权重值相同的 2 个感知机向量.由于二叉树在训练多次的判别中选取最优划分,而多叉树在二分类数据集上训练出的实际是一个判别式,所以二叉树存在微小的准确率优势.由表 2 可知:在 Breast 数据集上,二者准确率差距为 0.1%;而在 Spambase 数据集上,差距为 0.36%.

表 2 二叉树和多叉树的性能对比表

Tab. 2 Performance comparison table for binary tree and multifork tree

数据集	多叉树				二叉树				准确率/%	
	平均准确率/%	准确率范围/%	标准差	变异系数/%	平均准确率/%	准确率范围/%	标准差	变异系数/%	CART	C4.5
Balance	89.34	87.36~90.56	0.008 2	0.009 2	91.06 * *	90.08~92.32	0.006 3	0.006 9	72.00	72.32
Breast	96.82	96.33~97.36	0.003 0	0.003 1	96.72	96.19~97.36	0.002 8	0.002 9	92.08	92.82
Glass	64.58	58.41~70.56	0.029 0	0.044 9	65.09	60.75~68.69	0.019 1	0.029 3	70.56	66.82
Iris	96.04	94.00~98.00	0.009 6	0.010 0	95.78	94.00~97.33	0.007 6	0.007 9	92.67	93.33
Spambase	91.68	91.15~92.13	0.002 6	0.002 8	92.04	91.44~92.70	0.002 8	0.003 0	89.02	87.11
Segment	94.32	93.67~95.24	0.003 3	0.003 5	95.17 * *	94.43~95.57	0.003 2	0.003 4	91.48	92.86
Sensor	98.10	97.76~98.44	0.001 8	0.001 8	99.91 * *	99.76~99.98	0.000 4	0.000 4	100.00	100.00
Wine	57.42	56.16~58.66	0.006 6	0.011 5	58.52	56.16~60.29	0.009 0	0.015 4	53.53	51.53

在剩余的 6 个多类别数据集中,Iris 数据集上二叉树的平均准确率比多叉树低了 0.26%,其余 5 个数据集二叉树的准确率较多叉树分别提升了 1.72%,0.51%,0.85%,1.81%,1.10%,这些提升均高于前面在二分类数据集上两种树结构的准确率差距.采用 T 检验法对这 6 个数据集进行单侧显著性检验,查表后认为:Glass,Iris,Wine 数据集中,二叉树和多叉树分类效果没有显著差异;在 Balance,Segment,Sensor 数据集中,二叉树的分类效果显著优于多叉树;在准确率最低的 Wine 数据集中,原始数据内部类别不平衡问题较为严重,生成的决策树往往不能区分样本数量少的类别,由此带来准确率低下的

问题.

在和经典决策树算法的对比时,有 6 个数据集在不同程度上优于 CART 和 C4.5,可以认为是多变量决策树相对于轴平行决策树的固有优势.对于复杂的分类边界,轴平行决策树只能用直角折线取代斜线.在 Sensor 数据集中,由 CART 和 C4.5 的结果可以看出,这是一个线性可分的数据集.查看原始数据并对比 CART 和 C4.5 的决策树结果后发现,此数据集在边界上存在个别距离极近的点,实验中设置的感知机步长相对边界点间的距离较大,导致口袋算法建立的决策树无法学习最优的边界,而多叉树在性能上更落后一些.在 Glass 数据集上,无论二叉树还是多叉树都明显劣于经典决策树,这是因为 Glass 数据集中平均每一类数据的数据量仅为 30,因此,难以训练出可靠稳定的结果.

时间复杂度上,CART 和 C4.5 为精确线性搜索,会对每个特征的每个可能值进行分割并统计准确率,以找到最优,可以近似认为其在搜索过程中和样本的比对次数为  $k \times n^2$ .口袋算法的时间复杂度主要与遍历轮数的阈值有关,在运行中,若数据集可达到的准确率较高,则触发口袋向量检验更新的次数就减少,再加上口袋算法是多变量的,所以在运行中感知机向量与样本的比对次数与特征数  $k$  无关.

各模型在各数据集上的运行时间  $t$ ,如表 3 所示.由表 3 可知:二叉树与多叉树由于基本理论近似,所以时间的量级相同.由于经典单变量决策树算法在边界搜索中是遍历式的精确搜索,但是对全数据集仅遍历一次,所以当数据集样本数和特征数均较小时,运算速度比口袋算法建立的决策树要快一些.而当数据的特征与样本数较大时,花费时间会随之增长,且增长速度比口袋算法的决策树快,可以认为,在速度上口袋算法构造的多变量决策树具有速度优势.

表 3 各数据集上的运行时间

Tab. 3 Running time on each data set

s

数据集	$t$ (二叉树)	$t$ (多叉树)	$t$ (CART)	$t$ (C4.5)
Balance	1.023 2	0.851 0	0.132 8	0.157 8
Breast	0.285 2	0.835 8	0.104 7	0.107 8
Glass	0.984 2	0.919 6	1.273 4	1.257 8
Iris	0.118 1	0.140 5	0.059 4	0.040 6
Spambase	2.755 1	5.682 1	332.980 0	235.289 1
Segment	7.789 5	4.225 2	157.840 0	143.009 4
Sensor	9.251 6	6.471 8	31.491 0	34.117 2
Wine	7.760 2	8.971 0	10.858 0	7.195 3

## 5 结论

建立多变量决策树的方法有很多,选择口袋算法建树的原因是相比于其他多变量决策树,口袋算法理论和计算式都较为简单、易于理解.早期文献虽然尝试采用口袋算法建立决策树,但是仅按照原算法应用于二分类的数据集中<sup>[20]</sup>.基于口袋算法构造的二叉树和多叉树在分类准确率和运算时间上,均优于传统的轴平行决策树,而二叉树在分类准确率上,又能小幅优于多叉树.在对数据的解释性上,传统的轴平行决策树在单个节点用单一变量划分空间划分的解释性最强,多变量二叉树用分类超平面划分可以直接表示样本空间的划分边界,解释性比传统决策树弱,而多变量多叉树决策向量不代表空间边界,因此,在空间划分的解释性上要劣于前二者.

适用性上,当数据集为线性可分数据时,基于口袋算法的二叉树和多叉树均未能达到 100% 的准确率.因此,在实际使用中,若分类结果接近 100%,应当考虑数据集线性可分的可能性.而当训练集中类别较多、每类样本较少时,如果测定的结果方差较大,应当考虑是否能够学习得到最优的决策树.数据预处理中,对于类别不平衡的问题,可以采用对小样本进行样本扩充以近似类别平衡,也可以采用一些聚类算法约简大类别样本的数据量,这些可以根据具体问题采取合适的方法.

## 参考文献:

[1] MUSELLI M. On convergence properties of pocket algorithm[J]. IEEE Transactions on Neural Networks,1997,8(3):623. DOI:10.1109/72.572101.

[2] YILDIZ O T,ALPAYDIN E. Linear discriminant trees[J]. International Journal of Pattern Recognition and Artificial Intelligence,2005,19(3):323-353.

[3] MURTHY S K,KASIF S,SALZBERG S. A system for induction of oblique decision trees[J]. Journal of Artificial Intelligence Research,1994,2(1):1-32.

- [4] LOMAX S,VADERA S. A survey of cost-sensitive decision tree induction algorithms[J]. *Acm Computing Surveys*, 2013,45(2):1-35. DOI:10.1145/2431211.2431215.
- [5] 苗夺谦,王珏. 基于粗糙集的多变量决策树构造方法[J]. *软件学报*,1997(6):425-431.
- [6] LIM T S,LOH W Y,SHIH Y S. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms[J]. *Machine Learning*,2000,40(3):203-228. DOI:10.1023/A:1007608224229.
- [7] UTGOFF P E,BRODLEY C E. *Linear machine decision trees*[R]. Amherst:University of Massachusetts,1991.
- [8] SAHU S K,PUJARI A K,KUMAR V,*et al.* Greedy partitioning based tree structured multiclass SVM for Odia OCR[C]// *Computer Vision, Pattern Recognition, Image Processing and Graphics*. Patna:IEEE Press,2016:1-4. DOI:10.1109/NCVPRIPG.2015.7490018.
- [9] XUE Song,JING Xiaojun,SUN Songlin,*et al.* Binary-decision-tree-based multiclass support vector machines[C]// *International Symposium on Communications and Information Technologies*. Inchon:IEEE Press,2014:85-89. DOI:10.1109/ISCIT.2014.7011875.
- [10] MOUSTAKIDIS S,MALLINIS G,KOUTSIAS N,*et al.* SVM-based fuzzy decision trees for classification of high spatial resolution remote sensing images[J]. *IEEE Transactions on Geoscience and Remote Sensing*,2012,50(1):149-169. DOI:10.1109/TGRS.2011.2159726.
- [11] RAILEANU L E,STOFFEL K. Theoretical comparison between the Gini index and information gain criteria[J]. *Annals of Mathematics and Artificial Intelligence*,2004,41(1):77-93.
- [12] 潘大胜,屈迟文. 一种改进 ID3 型决策树挖掘算法[J]. *华侨大学学报(自然科学版)*,2016,37(1):71-77. DOI:10.11830/ISSN.1000-5013.2016.01.0071.
- [13] BREIMAN L,FRIEDMAN J H,OLSHEN R,*et al.* Classification and regression trees[J]. *Biometrics*,1984,40(3):358. DOI:10.2307/2530946.
- [14] GRAJSKI K A,BREIMAN L,VIANA D P G,*et al.* Classification of EEG spatial patterns with a tree-structured methodology: CART[J]. *IEEE Transactions on Biomedical Engineering*,1986,33(12):1076-1086. DOI:10.1109/TBME.1986.325684.
- [15] DE'ATH G,FABRICIUS K E. Classification and regression trees: A powerful yet simple technique for ecological data analysis[J]. *Ecology*,2000,81(11):3178-3192. DOI:10.2307/177409.
- [16] QUINLAN J R. C4. 5: Programs for machine learning[M]. San Francisco:Morgan Kaufmann Publishers Inc,1992.
- [17] MINGERS J. An empirical comparison of pruning methods for decision tree induction[J]. *Machine Learning*,1989,4(2):227-243. DOI:10.1023/A:1022604100933.
- [18] MINGERS J. An empirical comparison of selection measures for decision-tree induction[J]. *Machine Learning*,1989,3(4):319-342. DOI:10.1007/BF00116837.
- [19] KOTSIANTIS S B. Decision trees: A recent overview[J]. *Artificial Intelligence Review*,2013,39(4):261-283. DOI:10.1007/s10462-011-9272-4.
- [20] BRODLEY C E,UTGOFF P E. Multivariate decision trees[J]. *Machine Learning*,1995,19(1):45-77. DOI:10.1023/A:1022607123649.

(责任编辑:黄晓楠 英文审校:吴逢铁)