文章编号:1006-0464(2019)01-0090-07

C4.5 算法的研究及改进

姜如霞,黄水源,段文影,余楚波

(南昌大学信息工程学院,江西 南昌 330031)

摘 要:C4.5 算法作为目前常用的数据挖掘方法,仍存在一些缺陷。针对算法中出现的信息增益率计算复杂的问题,通过数学知识对增益率计算过程进行简化,提高计算效率;针对算法中可能偏袒属性值较多的属性的不足,在非类属性进行最佳属性的选择时引入权重这个概念;针对连续属性离散化过程耗时的缺陷,利用边界定理寻找最大信息增益率的候选分裂点,减少计算时间。将改进后的算法应用到葡萄牙某银行挖掘认购存款的潜在用户上,实验结果表明,C4.5 改进算法计算量减少,分类准确率也有提高,决策树的生成时间也大大缩减,构建的决策树贴合实际。

关键词:C4.5 算法;数学;权重系数;连续属性;边界定理

中图分类号:TP391 文献标志码:A

DOI:10.13764/j.cnki.ncdl.2019.01.017

A research and improvement of C4.5 algorithm

JIANG Ruxia, HUANG Shuiyuan, DUAN Wenying, YU Chubo

(School of Information Engineering, Nanchang University, Nanchang 330031, China)

Abstract: C4.5 algorithm is a frequently-used data mining method, but it still has some disadvantages. In order to solving the problem of the complexity of the information gain rate in the algorithm, the calculation process of gain rate is simplified by using mathematical knowledge, and the computational efficiency is improved. The weight coefficient is introduced when choosing the best attribute of the non-class attribute in the algorithm to overcome defects that attributes that may favor more attribute values Aiming at the shortcoming of the continuous attribute discretization process, the boundary theorem is used to find the candidate splitting points of maximum information gain rate, it reduces the computational time. The improved algorithm is applied to finding potential users who subscribe for deposits at a bank in Portugal. The experimental results show that the calculation of C4.5 is reduced, the generation time of decision tree is greatly reduced, the classification accuracy is improved and the decision tree satisfies the actual situation.

Key words: C4.5 algorithm; math; weight coefficient; continuous attribute; boundary theorem

在全球信息化大潮的推动下,数据挖掘的应用日趋广泛。许多决策者利用数据挖掘技术从海量的数据中获取有用的信息,为更好的决策提供帮助。基于此种需求,导致了数据挖掘研究和应用的蓬勃发展。决策树算法作为应用最广泛的数据挖掘方法之一,它的研究及改进一直在进行中。决策树(decision tree)算法基于特征属性进行分类,其主要的优点:模型具有可读性,计算量小,分类速度快。本质上决策树是通过一系列规则对数据进行分类的过程。构造决策树的常用算法包括了由 Quinlan 提出的 ID3 与 C4.5,Breiman 等提出的 CART。其中,C4.5 算法是在 ID3 算法的基础上,对于树的偏倚、连续属性离散化处理、缺值处理等问题进行了改进,从而使之具有更好的适应性。然而在传统的 C4.5 算法中,树的构造是按照深度优先策略完成的,需要对每个属性列表在每个结点处都进行一遍扫描,因

收稿日期:2018-06-26。
基金项目:国家自然科学基金资助项目(61070139,81460769)。
作者简介:姜如霞(1993—),女,硕士生。*通信作者:黄水源(1979—),男,副教授,硕士。E-mail:19111650@qq.com。

而导致算法的低效以及过度分支等问题。

目前越来越多的学者对 C4.5 算法进行研究改进并应用到不同的领域。文献[1]针对连续值属性离散化处理过程中耗时的缺点,使用 Fayyad 边界定理简化算法,并应用于金融借贷数据。文献[2]运用泰勒级数和等价无穷小的原理对算法的计算公式进行简化,同时引入其他非类属性对该属性的 gini 指数的均值,用于调整因非类属性间冗余度问题导致的误差。文献[3]引进新的参数 K,调整属性度量标准信息增益率的取值范围,进而应用到期货数据。

针对信息增益率计算复杂、容易出现过拟合和生成树的偏倚等问题,本文通过简化信息增益率的计算,非类属性进行最佳属性的选择时增加权重系数,以及采用 Fayyad 边界原理简化连续属性离散化算法等方法对 C4.5 算法进行改进。

1　算法分析

决策树是指用来表示决策和相应的决策结果对应关系的树。树中每一个非叶结点表示一个决策,该决策的值导致不同的决策结果(叶节点)或者影响后面的决策决策。本质上决策树是通过一系列规则对数据进行分类的过程。构造决策树的常用算法有 ID3、C4.5 和 CART 等。C4.5 算法用信息增益率来选择决策属性,其核心算法是 ID3 算法。它继承了 ID3 的全部优点,并在 ID3 的基础上增加了对连续属性的离散化、对未知属性的处理和产生规则等功能,克服了 ID3 算法的不足。

设数据分区 D 为标记类元组的训练集。假定类标号属性有 m 个不同的值,定义 m 个不同类 C_i 其中 i 的取值范围为 $\{1,2,3\cdots,m\}$。设 $C_{i,D}$ 是 D 中 C_i 类元组的集合,$|D|$ 和 $|C_{i,D}|$ 分别表示 D 和 $C_{i,D}$ 中元组的个数。对 D 中的元组分类所需要的期望信息由下式给出:

$$\mathrm{Info}(D)=-\sum_{i=1}^{m}p_i\log_2(p_i) \tag{1}$$

其中,p_i 是 D 中任意元组属于类 C_i 的非零概率,并用 $|C_{i,D}|/|D|$ 估计。

设属性 A 是离散值的,且属性 A 将 D 划分为 v 个分区或子集 $D_1,D_2,\cdots,D_v$,其中,D_j 包含 D 中的元组,它们的 A 值为 a_j。这些分区对应于从结点 N 生长出来的分支。根据由 A 划分成子集的期望信息由下式给出:

$$\mathrm{Info}_A(D)=\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times\mathrm{Info}(D_j) \tag{2}$$

其中,$|D_j|/|D|$ 充当第 j 个分区的权重。$\mathrm{Info}_A(D)$ 的值越小,分区的纯度越高。

假设属性 A 是连续值的,此时必须确定 A 的最佳分裂点,其中分裂点是 A 上的阈值。首先,将 A 的值按递增序排序。每对相邻值的中点作为可能的分裂点。给定的 A 的 v 个值,则需要计算 $v-1$ 个可能的划分。例如,A 的值 a_i 和 a_{i+1} 之间的中点是 $\frac{a_i+a_{i+1}}{2}$。

对于 A 的每一个可能分裂点,计算 $\mathrm{Info}_A(D)$,其中分区的个数为 2,即 $v=2$。A 具有最小期望信息需求的点选做 A 的分裂点。D_1 是满足 $A<=\mathrm{Splitinfo}$ 的元组集合,而 D_2 是满足 $A>\mathrm{Splitinfo}$ 的元组集合。

信息增益是原来的信息需求(仅基于类比例)与新的信息需求(对 A 划分后)之间的差。即衡量数据变纯的程度:

$$\mathrm{Gain}(A)=\mathrm{Info}(D)-\mathrm{Info}_A(D) \tag{3}$$

$\mathrm{Splitinfo}_A(D)$ 是数据集 D 划分成对应于属性 A 的 v 个输出的 v 个分区产生的信息。$\mathrm{GainRatio}(A)$ 是数据集 D 根据属性 A 划分的增益率:

$$\mathrm{Splitinfo}_A(D)=-\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times\log_2\left(\frac{|D_j|}{|D|}\right) \tag{4}$$

$$\mathrm{GainRate}(A)=\frac{\mathrm{Gain}(A)}{\mathrm{Splitinfo}_A(D)} \tag{5}$$

通过以上公式计算各个非类属性的信息增益率,依次选择具有最大增益率的属性作为分裂属性。信息增益率表示了由分支产生的有用信息的比率。因此,这个值越大,分支包含的有用的信息越多。

2　算法改进

C4.5 算法比一些其它分类模型易于理解,模型推出的规则有非常直观的解释,面对数据遗漏和输入字段很多的问题时非常稳健。但 C4.5 算法仍然存在一些缺陷,具体表现在可能偏袒属性值较多的属性、信息增益率计算复杂、连续属性离散化过程耗时。

2.1　引入权重系数

C4.5 算法会偏袒属性值数目较多的属性,在这

种情况下,将趋向于丢弃数据量少的数据元素,但是属性值较多的属性不总是最佳的属性。针对此种情况进行改进,即在计算信息熵时使用特尔斐法引入权重系数,来区分不同非类属性的重要程度。

专家咨询权树法(特尔斐法)主要根据专家对指标的重要性打分来确定指标的权重,分值越高,权重越大,重要性越高。这种方法的优点是集中了众多专家的意见,缺点是指标的权重具有很强的主观性。在实际的数据挖掘应用中,多个领域专家可以给定合理的取值。

设因素集 $A=\{a_1,a_2,\cdots,a_n\}$,k 个专家,每个专家独立给出因素 a_j 的权重。下面为 k 个专家给出所有因素的权重排成矩阵:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k1} & a_{k2} & \cdots & a_{kn} \end{pmatrix}$$

权重取加权平均:

$$a_j=\frac{1}{k}\sum_{i=1}^{k}a_{ij}(j=1,2,\cdots,n)$$

即得权重集 $A=(a_1,a_2,\cdots a_n)$。

2.2 简化信息增益率的计算

在计算信息增益率过程中涉及到对数函数的计算,所以在计算程序中就要调用库函数,同时随着数据量的增大,计算量也随之增大,这样就增加了计算时间,所以根据上述信息量计算公式的特点,转换为一种新的计算形式,使其去库函数中调用对数函数减少,加快算法生成树的速度。

根据数学公式的 log 换底变换以及等价无穷小的理论可以把式(1)转化成式(6);式(2)中若属性 A 为离散属性转化为式(7),若属性 A 为连续属性转化为式(8);式(3)转化成式(9);式(4)中若属性 A 为离散属性转化为式(10),若属性 A 为连续属性转化为式(11)。同时在计算按照属性 A 划分数据集的信息熵时引入权重系数。

$$\mathrm{Info}'(D)=-\sum_{i=1}^{m}\frac{|C_{i,D}|}{|D|}\log_2\left(\frac{|C_{i,D}|}{|D|}\right)=-\sum_{i=1}^{m}\frac{|C_{i,D}|}{|D|}\frac{\ln\frac{|C_{i,D}|}{|D|}}{\ln 2}=\frac{1}{|D|\ln 2}\sum_{i=1}^{m}\frac{|C_{i,D}|\times(|D|-|C_{i,D}|)}{|D|} \tag{6}$$

$$\mathrm{Info}'_A(D)=a_k-\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times\mathrm{Info}'(D_j)=a_k\sum_{j=1}^{v}\times\left(-\sum_{i=1}^{m}\frac{|C_{i,D_j}|}{|D_j|}\times\frac{\ln\frac{|C_{i,D_j}|}{|D_j|}}{\ln 2}\right)=\frac{a_k}{|D|\ln 2}\sum_{j=1}^{v}\sum_{i=1}^{m}\frac{C_{i,D_j}|\times(|D_j|-|C_{i,D_j}|)}{|D_j|} \tag{7}$$

$$\mathrm{Info}'_A(D)=a_k\left(\frac{|D_{\leqslant\mathrm{split}}|}{|D|}\times\mathrm{Info}'(|D_{\leqslant\mathrm{split}})+\frac{|D_{>\mathrm{split}}|}{|D|}\times\mathrm{Info}'(|D_{>\mathrm{split}})\right)=\frac{a_k}{|D|\ln 2}\left(-\sum_{i=1}^{m}\frac{|C_{i,D\leqslant\mathrm{split}}|\times(D_{\leqslant\mathrm{split}}|-|C_{i,D\leqslant\mathrm{split}}|)}{|D_{\leqslant\mathrm{split}}|}-\sum_{i=1}^{m}\frac{|C_{i,D>\mathrm{split}}|\times(|D_{>\mathrm{split}}|-|C_{i,D>\mathrm{split}}|)}{|D_{>\mathrm{split}}|}\right) \tag{8}$$

$$\mathrm{Splitinfo}'_A(D)=-\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times\log_2\left(\frac{|D_j|}{|D|}\right)=-\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times\frac{\ln\frac{|D_j|}{|D|}}{\ln 2}=\frac{1}{|D|\ln 2}\sum_{i=1}^{v}\frac{|D_j|\times(|D|-|D_j|)}{|D|} \tag{9}$$

$$\mathrm{GainRate}'(A)=\frac{\mathrm{Info}'(D)-\mathrm{Info}'_A(D)}{\mathrm{Splitinfo}_A(D)}=\frac{\sum_{i=1}^{m}\frac{|C_{i,D}|\times(|D|-|C_{i,D}|)}{|D|}-a_k\sum_{j=1}^{v}\sum_{i=1}^{m}\frac{|C_{i,D_j}|\times(|D_j|-|C_{i,D_j}|)}{|D_j|}}{\sum_{i=1}^{v}\frac{|D_j|\times(|D|-|D_j|)}{|D|}} \tag{10}$$

$$\mathrm{GainRate}'(A)=\frac{\mathrm{Info}'(D)-\mathrm{Info}'_A(D)}{\mathrm{Splitinfo}_A(D)}=$$

$$\frac{\sum_{i=1}^{m}\frac{|C_{i,D}|\times(|D|-|C_{i,D}|)}{|D|}-a_k\left(-\sum_{i=1}^{m}\frac{|C_{i,D\leqslant\text{split}}|\times(|D_{\leqslant\text{split}}|-|C_{i,D\leqslant\text{split}}|)}{|D_{\leqslant\text{split}}|}-\sum_{i=1}^{m}\frac{|C_{i,D\leqslant\text{split}}|\times(|D_{>\text{split}}|-|C_{i,D>\text{split}}|)}{|D_{>\text{split}}|}\right)}{\sum_{i=1}^{v}\frac{|D_j|\times(|D|-|C_j|)}{|D|}} \tag{11}$$

2.3　连续属性离散化算法的改进

C4.5 算法在连续属性离散化过程中,对所有相邻中点划分情况进行计算信息增益率,这将占用很多时间。针对上述问题,本文利用边界定理寻找候选分裂点,之后选择最大信息增益率的候选分裂点。

定义　属性 A 中的一个值 T 是一边界点,当且仅当在按 A 的值排序的实例集 S 中,存在两个实例 $e_1,e_2\in S$ 具有不同的类,使得 $A(e_1)<T<A(e_2)$,且不存在任何其他的实例 $e\in S$,使得 $A(e_1)<A(e)<A(e_2)$。$A(e)$ 表示实例 e 的属性 A 的取值。

Fayyad 和 Irani 定理:若 T 使得 $E(A,T,S)$ 最小,则 T 是一个边界点。其中 A 为属性,S 为实例集合,T 为某一阈值点。该定理表明,对连续属性 A,使得实例集合的平均类熵达到最小值的 T,总是处于实例序列中两个相邻异类实例之间。

连续属性离散化的最佳分界点必定出现在边界处。无需检查每一个阈值点,只要检查相邻不同类别的边界点即可。根据文献[1]的思想,以对 D_j 数据集排序为例,先按照其连续属性数值的大小对数据集进行排序,之后找出记录中类标号发生变化的点,最后计算这些点两边的连续属性平均值,把这些平均值作为边界点。在最好的情况下,按照连续属性排序后,各个记录按照其类标号刚好集中在一起,此时只有一个分界点;在最差的情况下,按照连续属性排序后,各个类标号都不同,此时分界点的个数为预测集数据总数减 1。因此用边界定理可以减少计算次数,提高计算效率。

2.4　算法改进描述

在构建决策树的过程中,改进的 C4.5 算法(C4.5_YH)首先需要采用特尔斐法计算权重,之后判断当前的属性是离散属性还是连续属性,若是离散属性按式(10)得到相应的信息增益率;若是连续属性,利用边界定理寻找属性的最佳分割阈值,再使用式(11)得到该属性的信息增益率。通过不同属性计算得出的信息增益率值的大小,确定具有最大信息增益率的属性为根节点。重复上述过程,完成决策树的构建。其相应的流程图如图 1 所示。

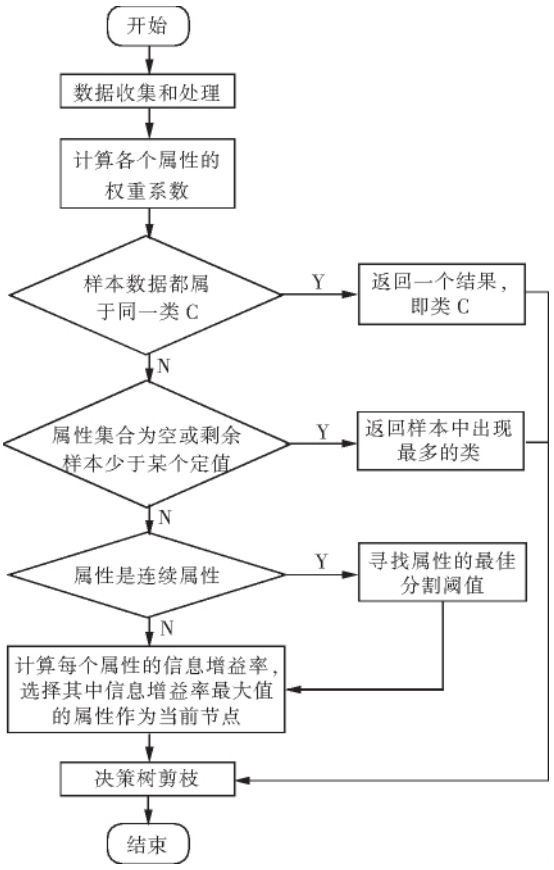

图 1　C4.5_YH 算法流程图

3　实验及结果分析

实验由两部分组成:(1)验证权重系数对最终决策树的影响;(2)C4.5_YH 算法在某银行挖掘潜在存款认购用户的应用。

3.1　实验 1

以一个典型的训练数据集 D(是否打球)表 1 为例。数据集有 4 个属性,即属性集合 $A=\{\text{weather},\text{temperature},\text{humidity},\text{windy}\}$。而类标签有 2 个,即类标签集合 $C=\{\text{yes},\text{no}\}$,分别表示适合户外运动和不适合户外运动。其中类属性值为 yes 的有 9 个样本,no 的样本有 5 个。

对样本数据用专家咨询权树法得到各个属性的权重值为 1、0.9、0.6、1,再根据 —— 计算 weather、temperature、humidity、windy 的信息增益率分别为 0.156 0、0.077 2、0.466 6、0.139 3。由此选择 humidity 作为根节点进行分类,重复过程得到最终的决策树。

表 1 训练数据集(是否打球)

weather	temperature	humidity	windy	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cold	normal	FALSE	yes
rainy	cold	normal	TRUE	no
overcast	cold	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cold	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

对 C4.5 算法所得的决策树(图 2)和 C4.5_YH 算法所得的决策树(图 3)进行比较,可以发现权重较大的属性离根节点距离远。根据比较结果,C4.5_YH算法所构造的决策树与实际情况较为符合,能达到预期目的。

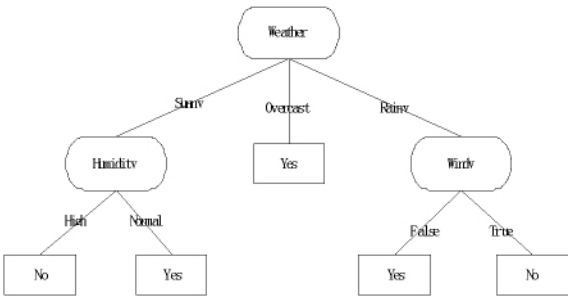

图 2 C4.5 算法生成的决策树

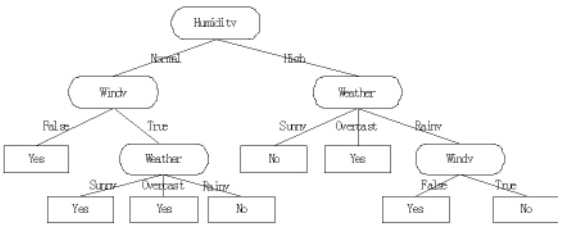

图 3 C4.5_YH 算法生成的决策树

3.2 实验 2

为比较 C4.5 算法与 C4.5_YH 算法的预测能力,使用葡萄牙某银行在 2008 年 5 月至 2010 年 11 月的存款认购历史数据进行验证。在怀卡托智能分析环境(Waikato Environment for Knowledge Analysis)下进行实验,weka 版本为 3.8.1。对历史数据进行分析,从中寻找到有价值的信息,以此指导葡萄牙某银行挖掘认购存款的潜在用户。

3.2.1 数据准备

存款认购历史数据集(Bank_Marketing)数据量有 4 万,包括 16 个属性,其中 7 个为连续属性。其中 goal 作为目标属性,其它则为条件属性,如下所示。表 1 列出处理后的部分数据集。

(1) age:存款人年龄。

(2) job:存款人职业(行政人员、企业家、工人、失业、退休、未知等)。

(3) marital:存款人婚姻情况。

(4) education:存款人教育情况。

(5) credit default:是否有信用违约。

(6) average yearly balance:年平均余额(欧元)。

(7) housing loan:是否存在房屋贷款。

(8) personal loan:是否存在个人贷款。

(9) communication type:联系人通信类型("未知","电话","蜂窝")。

(10) day:最后联络时间日期。

(11) month:最后联络时间月份。

(12) duration:通话时间(单位:秒)。

(13) contacts:此次活动中联系次数。

(14) pdays:上次活动与此次活动的间隔天数(-1 表示之前从未联系)。

(15) previous:此次活动之前联系次数。

(16) poutcome:以往的营销活动的结果。("未知的"、"其他"、"失败"、"成功")

(17) goal:是否会认购此次活动银行定期存款,yes 表示会,no 表示不会。

3.2.2 算法检验

把 C4.5_YH 算法加入 weka 软件中,使用它对存款认购历史数据集(Bank_Marketing)进行处理得到分类模型。图 4 为其中部分模型。

```
duration <= 410
|   poutcome = unknown
|   |   age <= 60
|   |   |   month = may: no (9338.0/101.0)
|   |   |   month = jun
|   |   |   |   communication type = unknown: no (3834.0/12.0)
|   |   |   |   communication type = cellular
|   |   |   |   |   duration <= 156: no (158.0/19.0)
|   |   |   |   |   duration > 156
|   |   |   |   |   |   job = management
|   |   |   |   |   |   |   education = tertiary
|   |   |   |   |   |   |   |   age <= 46
|   |   |   |   |   |   |   |   |   average yearly balance <= 415: no (9.0/1.0)
|   |   |   |   |   |   |   |   |   average yearly balance > 415
|   |   |   |   |   |   |   |   |   |   average yearly balance <= 1377: yes (12.0/2.0)
|   |   |   |   |   |   |   |   |   |   average yearly balance > 1377
|   |   |   |   |   |   |   |   |   |   |   marital = married
|   |   |   |   |   |   |   |   |   |   |   |   age <= 41: no (6.0)
|   |   |   |   |   |   |   |   |   |   |   |   age > 41: yes (3.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   marital = single
|   |   |   |   |   |   |   |   |   |   |   |   average yearly balance <= 2863: no (7.0)
|   |   |   |   |   |   |   |   |   |   |   |   average yearly balance > 2863: yes (6.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   marital = divorced: no (2.0/1.0)
|   |   |   |   |   |   |   |   age > 46: yes (13.0/2.0)
|   |   |   |   |   |   |   education = secondary: yes (3.0/1.0)
|   |   |   |   |   |   |   education = unknown: yes (0.0)
|   |   |   |   |   |   |   education = primary: yes (1.0)
|   |   |   |   |   |   job = technician
|   |   |   |   |   |   |   personal loan = no
```

图 4 部分预测模型

3.2.3 算法结果分析

进入 weka 的实验界面,添加数据集和相关算

法。实验采用 10 重交叉验证的方法,对具有相同属性个数,相同数量的数据集(Bank_Marketing)分别进行 C4.5 算法, C4.5_YH 算法与 NaiveBayes 的对比实验,得到结果见表 2。Correct 表示算法分类准确率,time 表示建立模型所用的时间。

```
=== Summary ===

Correctly Classified Instances       40831               90.3121 %
Incorrectly Classified Instances      4380                9.6879 %
Kappa statistic                          0.4839
Mean absolute error                      0.1269
Root mean squared error                  0.2773
Relative absolute error                 61.4259 %
Root relative squared error             86.2833 %
Total Number of Instances            45211
```

图 5　C4.5_YH 算法实验结果

```
=== Summary ===

Correctly Classified Instances       40791               90.2236 %
Incorrectly Classified Instances      4420                9.7764 %
Kappa statistic                          0.4732
Mean absolute error                      0.1288
Root mean squared error                  0.2682
Relative absolute error                 62.3586 %
Root relative squared error             83.4513 %
Total Number of Instances            45211
```

图 6　C4.5 算法实验结果

```
=== Summary ===

Correctly Classified Instances       39789               88.0073 %
Incorrectly Classified Instances      5422                11.9927 %
Kappa statistic                          0.4391
Mean absolute error                      0.1532
Root mean squared error                  0.3088
Relative absolute error                 74.1603 %
Root relative squared error             96.0681 %
Total Number of Instances            45211
```

图 7　NaiveBayes 算法实验结果

表 2　部分存款认购人数据表

id	age	job	marital	education	credit default	average yearly balance	…	goal
1	30	unemployed	married	primary	no	1 787	…	no
2	33	services	married	secondary	no	4 789	…	no
3	35	management	single	tertiary	no	1 350	…	no
4	30	management	married	tertiary	no	1 476	…	no
5	59	blue-collar	married	secondary	no	0	…	no
6	35	management	single	tertiary	no	747	…	no
7	36	self-employed	married	tertiary	no	307	…	no
8	39	technician	married	secondary	no	147	…	no
9	41	entrepreneur	married	tertiary	no	221	…	no
10	43	services	married	primary	no	−88	…	no
11	39	services	married	secondary	no	9 374	…	no
…	…	…	…	…	…	…	…	…

表 3　算法结果比较表

dataset	Algorithm	Correct/%	time/s
Bank_Marketing	C4.5_YH	90.312 1	0.64
	C4.5	90.223 6	1.82
	NaiveBayes	88.007 3	0.82

由表 3 和图 4－7 可知,C4.5_YH 算法建立的预测模型在三个模型中准确性最高为 90.3121%。在耗时上也比 C4.5 算法和 NaiveBayes 算法低,大大提高决策树的生成效率。由此,在葡萄牙某银行挖掘认购存款的潜在用户的实际应用中, C4.5_YH 算法是具有指导性作用,有一定的实际意义。

4　结语

C4.5 作为经典的决策树算法,有着直观、效率高的优点,因而得到广泛的应用。本文针对 C4.5 算法中的不足加以改进,使用对数变换及等价无穷小等数学知识化简信息增益率,以各个属性的权重系数调整算法对多属性值的偏倚,并以边界定理简化连续属性离散化算法,在缩短建模时间的同时,也提高了准确率。本文提出 C4.5_YH 算法未能对剪枝方面的改进进行研究,这些将在后续的研究工作中完成。

参考文献:

[1] 苗煜飞,张霄宏.决策树 C4.5 算法的优化与应用[J].计算机工程与应用,2015,51(13):255-258.

[2] 黄秀霞,孙力.C4.5 算法的优化[J].计算机工程与设计,2016,37(5):1265-1270.

[3] 陈磊,何国辉.改进的 C4.5 算法在期货数据挖掘中的研究[J].计算机工程与应用,2017,53(11):161-166.

[4] 罗丽娟,段隆振,段文影,等.C5.0 算法的改进及应用[J].南昌大学学报(理科版),2017,41(1):92-97.

[5] 刘承启,黄学坚,许健锋,等.基于决策树和粗糙集的高分辨率短时临近雷电预报模型[J].南昌大学学报(理科版),2014,38(6):559-568.

[6] 姚亚夫,邢留涛.决策树 C4.5 连续属性分割阈值算法改进及其应用[J].中南大学学报(自然科学版),2011,42(12):3772-3776.

[7] ROIGER R J,GEATZ M W.数据挖掘教程[M].翁敬农,译.北京:清华大学出版社,2008:60-85.

[8] 李孝伟,陈福才,李邵梅.基于分类规则的 C4.5 决策树改进算法[J].计算机工程与设计,2013,34(12):4321-4325.

[9] 黄爱辉.决策树 C4.5 算法的改进及应用[J].科学技术与工程,2009,9(1):34-36.

[10] 魏浩,丁要军.一种基于属性相关的 C4.5 决策树改进算法[J].中北大学学报(自然科学版),2014,35(4):402-406.

[11] 谢妞妞,刘於勋.决策树属性选择标准的改进[J].计算机工程与设计,2010,46(34):115-118.

[12] HAN J,MICKELINE K,PEI J.数据挖掘:概念与技术[M].范明,孟小峰,译.北京:机械工业出版社,2012.

[13] WITTEN I H,FRANK E.数据挖掘实用机器学习技术[M].麦琳,邱泉,于晓峰,等译.北京:机械工业出版社,2006.

[14] 王苗,柴瑞敏.一种改进的决策树分类属性选择算法[J].计算机工程与应用,2010,46(8):127-129.