C4.5 算法的优化

黄秀霞，孙 力+

（江南大学 物联网工程学院，江苏 无锡 214112）

摘 要：对传统 C4.5 算法的运算效率和属性选择准确性进行研究，对其进行改进。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，提高运算效率；在简化后的信息增益率计算公式中引入其它非类属性对于该属性的 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，提高算法属性选择的准确性，将改进后的算法称为 G_C4.5。对 G_C4.5、传统 C4.5 算法与其它改进算法进行对比实验分析，分析结果表明，G_C4.5 算法在分类效率和准确性上都有一定提高。

关键词：C4.5 算法；泰勒级数；等价无穷小；GINI 指数的均值；非类属性间关联性；G_C4.5 算法

中图法分类号：TP311.5 **文献标识号**：A **文章编号**：1000-7024（2016）05-1265-06

doi：10.16208/j.issn1000-7024.2016.05.029

Optimization of C4.5 algorithm

HUANG Xiu-xia, SUN Li+

(School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China)

Abstract: After researching the computing efficiency and attribute selection accuracy of traditional C4.5 algorithm, some improvements were implemented. The calculation formula was simplified using the principle of Taylor series and equivalent infinitesimal, the efficiency of calculation was improved. The average value of GINI index of non-class attributes for this attribute was introduced to the simplified formula of information gain rate, the deviation caused by the redundancy between non-class attributes was adjusted, and the accuracy of the attribute selection was improved. The improved algorithm was named as G_C4.5. G_C4.5 algorithm was contrasted with traditional C4.5 algorithm and its other improved algorithms, results show that G_C4.5 algorithm improves the classification efficiency and the classification accuracy.

Key words: C4.5 algorithm; Taylor series; equivalent infinitesimal; average of GINI index; correlation between non-class attributes; G_C4.5 algorithm

0 引 言

数据挖掘有很多经典算法，J Ross Quinlan 提出一种分类预测算法——ID3 算法，并在此基础上进行改进优化，得到了 C4.5 算法。然而，传统的 C4.5 算法运算效率不高且属性选择上没有考虑到非类属性间相关性的影响，针对这些问题，对 C4.5 算法进行改进。主要做了以下优化：简化对数运算以提高计算效率；加入非类属性对于该属性的 GINI 指数的均值，得到新的信息增益率计算公式；为了消除简化带来的误差，引入每个属性的属性值个数[1]，得到最终的信息增益率公式。为验证其先进性，运用该算法对学校的“英语统考成绩”数据进行挖掘分析实验，同时对 G_C4.5 算法性能进行对比实验。

1 C4.5 算法的描述

数据挖掘[2,3]就是通过分析大量的数据，从中揭示出隐含的、未知的和具有潜在价值的信息的一个过程。数据挖掘常用的方法有：分类、回归分析、聚类、关联规则和特征分析等。常用的技术有：人工神经网络、决策树和遗传算法等。知识发现过程中关键的一步就是数据挖掘，知识发现的流程为：

（1）将数据进行清洗，从而消除干扰性的数据。

（2）将清洗好的多个数据集都集合在一起。

（3）根据要挖掘的内容进行筛选数据。

（4）将不同形式的数据集转变成适合挖掘的统一数据形式。

收稿日期：2015-06-13；**修订日期**：2015-08-17

作者简介：黄秀霞（1990-），女，广东梅州人，硕士研究生，CCF 会员，研究方向为计算机软件与理论；+通讯作者：孙力（1966-），男，江苏无锡人，博士，教授，研究方向为计算机科学与技术。E-mail：xiuxia_huang@126.com

(5) 对选定的数据进行挖掘分析(关键流程,用相应的挖掘算法提取所需信息)。

(6) 挖掘发现出来的模式经过评估,如若模式不合格则再退回第(3)步执行;合格则用直观的可视化的表示方法来向用户描述挖掘出来的相关知识。

构造决策树来对大数据进行分析处理是现在数据挖掘技术的普遍方法。构造决策树的算法有很多,包括 ID3、C4.5 和 CART 等都是常用的决策树算法。

ID3 算法[4]是 J Ross Quinlan 提出的一种分类预测算法,该算法的核心是信息熵和信息增益。信息熵在物理学上是描述信源的不确定度的,而在数学上则表示了信息冗余度和概率之间的关系。两个信息熵的差值就是信息增益度。数据集中样本对于一个类的信息熵其中一个信息熵的值;而另一个信息熵已知一个属性的值后这个数据集中的样本关于一个类的信息熵。对数据集中的每一个属性运用 ID3 算法中的信息增益公式进行计算,得到一系列的值,选取其中最大的值并将其对应的属性作为当前数据集的测试属性。如果用决策树来描述数据集,节点则为经过计算选出的测试属性,再以这个属性为标记,它的每个值都是它的一个分支,以此类推来划分数据集中的样本。

C4.5 算法[5]是在 ID3 算法基础上通过几个方面的完善得到的。最关键的改进是利用信息增益率来弥补了 ID3 算法中用信息熵作为选择分支属性标准的不足,即在 ID3 算法中,属性选择时会偏向于选择属性取值多的属性。同时 C4.5 也弥补了 ID3 算法中缺少对空缺值的处理和连续属性离散化处理的缺陷。

C4.5 算法的主要思想:

设 T 为类标记元组的训练集。设 $C_i(i=1,2,\cdots,n)$ 为类标记属性具有 n 个不同的值。设 TC_i 是训练集 T 中属于 C_i 类的元组集合,而 $|T|$ 和 $|TC_i|$ 分别是训练集 T 和训练集 TC_i 中的元组个数;再把训练集 T 中的元组按属性 A 来划分,具有 m 个不同值为 $\{a_1,a_2,\cdots\cdots,a_m\}$,如果 A 是离散型的,则值与 A 上测试的 V 个输出直接对应,若 A 是连续性的则需要进行离散化处理。属性 A 将训练集 T 分为 m 个子集 $\{T_1,T_2,\cdots,T_m\}$,其中 T_j 包含了 T 中的元组,在属性 A 上的值为 a_j。主要计算:

(1) T 中元组的信息熵为

$$Info(T)=-\sum_{i=1}^{n}\frac{|TC_i|}{|T|}\log_2\frac{|TC_i|}{|T|} \tag{1}$$

(2) 按 A 划分 T 的元组分类的信息熵为

$$Info(AT)=\sum_{j=1}^{m}\frac{|T_j|}{|T|}Info(T_j) \tag{2}$$

(3) 属性 A 的信息增益

$$Gain(A)=Info(T)-Info(AT) \tag{3}$$

(4) 属性 A 的分裂信息为

$$SplitInfo(AT)=-\sum_{j=1}^{m}\frac{|T_j|}{|T|}\log_2\frac{|T_j|}{|T|} \tag{4}$$

(5) 属性 A 的信息增益率

$$GainRatio(A)=\frac{Gain(A)}{SplitInfo(AT)} \tag{5}$$

2 优化 C4.5 算法

2.1 提高计算效率

2.1.1 泰勒级数

在数学中,泰勒级数[6](Taylor series)是用无限项的连加式,即级数来表示一个函数,这些相加的项由函数在某一点的导数求得。其定义如下:一个在 a(a 为实数或复数)邻域上的无穷可微实变函数或复变函数 $f(x)$ 的泰勒级数是如下的幂级数

$$f(x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(x-a)^n \tag{6}$$

式中:n!——n 的阶乘;$f^{(n)}(a)$——函数 f 在点 a 处的 n 阶导数。如果 $a=0$,这个级数也被称为麦克劳伦级数。

根据定义,当 $f(x)$ 为自然对数时,即 $f(x)=\ln(x)$ 时,其泰勒级数为

$$\ln(1+x)=\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{n}x^n \tag{7}$$

根据等价无穷小原理,当 X 的值很小时,可得 $\sum_{n=2}^{\infty}\frac{(-1)^{n+1}}{n}x^n$ 的值都趋近于 0,即有

$$\ln(1+x)\approx x \tag{8}$$

2.1.2 简化计算

在传统 C4.5 的信息熵的计算中由于 $\log_2\frac{|TC_i|}{|T|}=\frac{\ln\frac{|TC_i|}{|T|}}{\ln 2}$,则根据式(8)原理进行简化得到

$$Info(T)=-\sum_{i=1}^{n}\frac{|TC_i|}{|T|}\log_2\frac{|TC_i|}{|T|}=-\frac{\ln\frac{|TC_i|}{|T|}}{\ln 2}\sum_{i=1}^{n}\frac{|TC_i|}{|T|}=\frac{1}{|T|\ln 2}\sum_{i=1}^{m}\frac{|TC_i|\times(|T|-|TC_i|)}{|T|} \tag{9}$$

同理,其它涉及对数运算的公式也进行相应的简化,最后得到信息增益率公式为

$$GainRatio(A)=\frac{Gain(A)}{SplitInfo(AT)}=\frac{\sum_{i=1}^{n}\frac{|TC_i|\times(|T|-|TC_i|)}{|T|}-\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{|TC_i|\times(|T_j|-|TC_j|)}{|T_j|}}{\sum_{j=1}^{m}\frac{|T_j|\times(|T|-|T_j|)}{|T|}} \tag{10}$$

式中：$|TC_{ij}|$即为类别 C_i 的元组中在属性 A 中的取值为 A_j 的元组个数。因为在化简的过程中会产生误差，所以不能直接运用式（10）来计算信息增益率。为弥补误差，加入每个属性的属性值个数 M[1]，即信息增益率分别乘以 M

$$GainRatioM(A)=\frac{Gain(A)}{SplitInfo(AT)}\times M \tag{11}$$

式（11）即为根据泰勒级数和等价无穷小的原理进行简化后的最终计算公式。这个信息增益率的计算公式不涉及对数运算，从而提高了算法的分类效率。

2.2 提高属性选择准确性

2.2.1 gini 指数

gini 指数[7]是一种不纯度函数（impurity function），已经用于一些分类算法，他是一种适合可分类型和数值型的分类的方法。不纯度函数可以计算数据集中的样本“纯度”。若在一个数据集中，样本集中地分布在某一个类中，则这个数据集的“纯度”就大。反之这个数据集的“纯度”就小。也就是说，将一个数据集根据相应的取值范围进行拆分，其“纯度”将减增大。其定义为：

如上面提到的数据集 T 包含 n 个类别的样本集，则其 gini 指数为

$$gini(T)=1-\sum_{i=1}^{n}\left(\frac{|TC_i|}{|T|}\right)^2 \tag{12}$$

如果某个属性 A 有 m 个取值，根据该属性的取值把集合 T 分为 m 个部分，$|T_j|$（$j=1,2,\cdots,m$）为属性 A 取第 j 个值时的样本个数，那么这个属性的 gini 指数为

$$giniSplit(AT)=\sum_{j=1}^{m}\left[\frac{|T_j|}{|T|}gini(T_j)\right] \tag{13}$$

2.2.2 运用 gini 指数改进算法

在决策树分类选择根属性时，信息增益率越高，说明该属性越纯净，则首先选择该属性进行分割。与此相反的，gini 指数主要是度量数据划分或训练数据集 D 的不纯度为主，如果计算的 gini 指数的值大，就表示数据集的不纯度高。即样本属于同一个类的概率就越高。这是属性与类属性之间的关系。

同样的在非类属性之间，如果一个属性与其它属性之间的信息增益率越大，则该属性与其它属性之间的相关性[8]就越大，即冗余度就越大；相反的，如果一个属性与其它属性间的 gini 指数越小，则它们之间的冗余度就越大。如式（14）所示即为一个属性（以属性 A 为例）与其它属性间的 gini 指数之和

$$Sum\_giniSplit(A_FT)=\sum_{i=1}^{s}\sum_{j=1}^{x}\left[\frac{|T_{ij}|}{|T|}gini(T_{ij})\right] \tag{14}$$

式中：A_FT 表示不包含类属性和属性 A 的属性集；s 表示除了类属性和属性 A 外的属性个数；x 是变量，表示每个非类属性（除属性 A 外）的属性值个数，即随着 i 的变化而变化；$|T_{ij}|$表示第 i 个非类属性（除属性 A 外）取第 j 个属性值时样本的个数。

$gini(T_{ij})$表示每个非类属性（除属性 A 外）关于属性 A 的 gini 指数，其计算如式（15）所示

$$gini(T_{ij})=1-\sum_{k=1}^{m}\left(\frac{|TA_{ijk}|}{|T_{ij}|}\right)^2 \tag{15}$$

式中：$|TA_{ijk}|$——第 i 个非类属性（除属性 A 外）取第 j 个属性值的同时，属性 A 为第 k 个值时的样本个数。

那么属性 A 与其它属性（不包括类属性）之间的 gini 指数之和的平均值为

$$\overline{Sum\_giniSplit(A_FT)}=\frac{\sum_{i=1}^{s}\sum_{j=1}^{x}\left[\frac{|T_{ij}|}{|T|}gini(T_{ij})\right]}{s} \tag{16}$$

为此，在计算属性信息增益率时，加入该属性与其它属性（不包括类属性）之间的 gini 指数的平均值来提高属性选择的正确性。即属性 A 的信息增益率的计算在运用泰勒级数和等价无穷小的原理进行简化后，属性 A 的分裂信息减去该属性与其它非类属性间的 gini 指数的均值，作为新的分裂信息，计算出信息增益率。如式（17）所示

$$GainR(A)=\frac{Gain(A)\times M}{SplitInfo(AT)-\overline{Sum\_giniSplit(A_FT)}}=\frac{\sum_{i=1}^{n}\frac{|TC_i|\times(|T|-|TC_i|)}{|T|}-\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{|TC_{ij}|\times(|T_j|-|TC_{ij}|)}{|T_j|}}{\sum_{j=1}^{m}\frac{|T_j|\times(|T|-|T_j|)}{|T|}-\overline{Sum\_giniSplit(A_FT)}}\times M \tag{17}$$

根据上式可知，如果属性 A 与其它非类属性间的相关性越小，即冗余度越小，那么属性 A 与其它非类属性间的 gini 指数平均值就越大，即 $\overline{Sum\_giniSplit(A_FT)}$ 的值就越大，相反的，$SplitInfo(AT)-\overline{Sum\_giniSplit(A_FT)}$ 的值就越小，则属性 A 的信息增益率越大。因此消除了非类属性间的冗余度对属性选择准确性的影响，提高了算法的分类准确性。

2.3 G_C4.5 算法描述

经过以上改进后的 C4.5 算法，我们称之为 G_C4.5 算法，与别的改进算法[9,10,12]不同的是，除了简化计算过程，还加入了非类属性间的 gini 指数平均值 $\overline{Sum\_giniSplit(A_FT)}$ 来提高属性选择的准确性。这是G_C4.5 算法的关键改进部分。

计算 $\overline{Sum\_giniSplit(A_FT)}$ 的伪代码具体描述为：

```
Function：AGSumGiniSplit()；
Begin
  For 所有的属性 R(Attribute(p)) Do {
    For 所有的属性 R(Attribute(i)) Do {
```

If 属性 Attribute(i) 不等于属性 Attribute(p)，并且不等于类属性，则

Begin

For 属性 Attribute(i) 的每一个值 (Attribute(i).numValues(j)) Do {

For 属性 Attribute(p) 的每一个值 (Attribute(p).numValues(k)) Do

计算 gini(T_{ij}) 的值赋给 Gi；

计算出属性 Attribute(i) 与其它非类属性 Attribute(p) 的 gini 指数赋给 GiS;}

End

计算出属性 Attribute(p) 与其它非类属性 Attribute(i) 的 gini 指数之和赋给 SumGiSp;}

将 SumGiSp 的均值依次赋给 (SGj/j=1，2…m);}

返回 SGj;

End

G_C4.5 算法流程描述如图 1 所示。

3 实验验证结果及分析

3.1 实验一

运用改进前 C4.5 算法和 G_C4.5 算法（改进后的算法）对学校英语统考成绩的数据进行实验对比，我们抽取全校各个专业的英语统考成绩信息建立数据表，进行数据的清理筛选，最终转换集成后得到下面的数据表（部分数据）。

表 1 统考成绩是否合格数据集

学号	性别	入学英语测试成绩	已学课程平均成绩	学习情况	英语统考情况
912931900	男	优	B	懒散	合格
912332811	女	优	A	一般	合格
913330517	男	中	C	懒散	不合格
913331924	女	中	C	一般	合格
912331263	女	优	B	一般	不合格
912930533	女	中	C	勤快	合格
912331668	女	良	B	勤快	合格
912332881	女	中	A	一般	合格
912932064	男	优	A	勤快	不合格
912932789	男	良	A	懒散	不合格
911931630	男	中	B	勤快	不合格
912332138	女	中	C	勤快	合格
912330019	男	良	B	一般	合格
912930615	女	优	A	懒散	不合格

从表 1 中，可以看到类属性“英语统考情况”有 2 个值，即数据被分为两类——合格和不合格。其中，类属性值为“合格”的有 8 个样本，“不合格”的样本有 6 个。

(1) 跟据改进前的 C4.5 算法计算出各个属性的信息增益率。

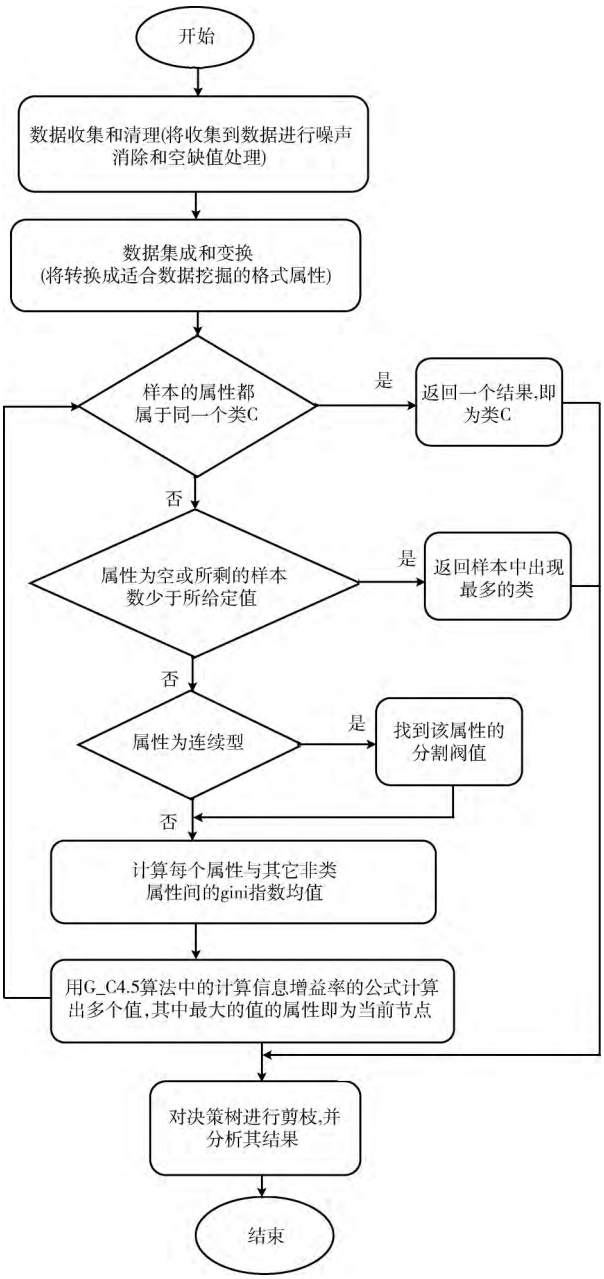

图 1 G_C4.5 算法流程

根据式 (1) 可以先计算出数据集的信息熵

$$Info(\text{统考情况}) = -\frac{8}{14}\times\log_2\frac{8}{14} - \frac{6}{14}\times\log_2\frac{6}{14} = 0.597613217$$

以属性“入学英语测试成绩”为例，其属性值一共有 3 个，分别为：“优”、“良”，“中”。属性值为“优”的样本有 5 个，其中 2 个类属性值为“合格”，3 个类属性值为“不合格”；属性值为“良”的样本有 3 个，其中 2 个类属性值为“合格”，1 个类属性值为“不合格”；属性值为“中”的样本有 6 个，其中 4 个类属性值为“合格”，2 个类属性值为“不合格”。则按“入学英语测试成绩”划分的数

据的信息熵为

$$Info(\text{入学英语成绩}) = \frac{5}{14}\times\left(-\frac{3}{5}\times\log_2\frac{3}{5}-\frac{2}{5}\times\log_2\frac{2}{5}\right)+\frac{3}{14}\times\left(-\frac{1}{3}\times\log_2\frac{1}{3}-\frac{2}{3}\times\log_2\frac{2}{3}\right)+\frac{6}{14}\times\left(-\frac{4}{6}\times\log_2\frac{4}{6}-\frac{2}{6}\times\log_2\frac{2}{6}\right)=0.583125536$$

$$GainRatio(\text{入学英语成绩}) = Info(\text{统考情况})-Info(\text{入学英语,统考情况}) = 0.014487681$$

$$SplitInfo(\text{入学英语成绩}) = -\frac{5}{14}\times\log_2\frac{5}{14}-\frac{3}{14}\times\log_2\frac{3}{14}-\frac{6}{14}\times\log_2\frac{6}{14}=0.761792225$$

$$GainRatio(\text{入学英语成绩}) = \frac{Gain(\text{入学英语成绩})}{SplitInfo(\text{入学英语成绩})} = \frac{0.014487681}{0.761792225} = 0.01901789$$

同理，可算出其它几个属性的信息增益率为

$$GainRatio(\text{性别}) = 0.064520171$$

$$GainRatio(\text{平时成绩}) = 0.023239739$$

$$GainRatio(\text{学习情况}) = 0.057745982$$

根据上面的计算结果，有：$GainRatio(\text{性别}) > GainRatio(\text{学习情况}) > GainRatio(\text{平时成绩}) > GainRatio(\text{入学英语成绩})$，即选择“性别”属性作为决策树的根节点。

(2) 根据改进后的 C4.5 算法计算出各个属性的信息增益率。

同样，以属性“入学英语测试成绩”为例，根据式(14)和式(15)计算出其它非类属性关于该属性的 gini 指数的总和。该属性的特征值个数为 3，即 $M=3$。由表 1 的数据可知：S 的值为 3。

当 $i=1$，属性为“性别”时，X 的取值为 2。其中属性值为“男”时，一共有 6 个样本，其“入学英语测试成绩”的取值分别为：2 个“优”、2 个“良”、2 个“中”；而属性值为“女”时，一有 8 个样本，具体取值为：3 个“优”、1 个“良”、4 个“中”。则属性“性别”关于“入学英语测试成绩”的 gini 指数为

$$giniSplit1(\text{入学英语成绩}_F T) = \frac{6}{14}\times\left\{1-\left[\left(\frac{2}{6}\right)^2+\left(\frac{2}{6}\right)^2+\left(\frac{2}{6}\right)^2\right]\right\}+\frac{8}{14}\times\left\{1-\left[\left(\frac{3}{8}\right)^2+\left(\frac{1}{8}\right)^2+\left(\frac{4}{8}\right)^2\right]\right\}=\frac{5}{8}$$

同理，当 $i=2$ 时，属性“已学课程平均成绩”关于“入学英语测试成绩”的 gini 指数为

$$giniSplit2(\text{入学英语成绩}_F T) = \frac{5}{14}\times\left\{1-\left[\left(\frac{3}{5}\right)^2+\left(\frac{1}{5}\right)^2+\left(\frac{1}{5}\right)^2\right]\right\}+\frac{5}{14}\times\left\{1-\left[\left(\frac{2}{5}\right)^2+\left(\frac{2}{5}\right)^2+\left(\frac{1}{5}\right)^2\right]\right\}+\frac{4}{14}\times\left\{1-\left[\left(\frac{4}{4}\right)^2\right]\right\}=\frac{3}{7}$$

当 $i=3$ 时，属性“学习情况”关于“入学英语测试成绩”的 gini 指数为

$$giniSplit3(\text{入学英语成绩}_F T) = \frac{5}{14}\times\left\{1-\left[\left(\frac{1}{5}\right)^2+\left(\frac{1}{5}\right)^2+\left(\frac{3}{5}\right)^2\right]\right\}+\frac{5}{14}\times\left\{1-\left[\left(\frac{2}{5}\right)^2+\left(\frac{1}{5}\right)^2+\left(\frac{2}{5}\right)^2\right]\right\}+\frac{4}{14}\times\left\{1-\left[\left(\frac{2}{4}\right)^2+\left(\frac{1}{4}\right)^2+\left(\frac{1}{4}\right)^2\right]\right\}=\frac{17}{28}$$

则属性“入学英语测试成绩”与其它属性（不包括类属性）之间的 gini 指数平均值为

$$\overline{Sum\_giniSplit(\text{入学英语成绩}_F T)} = \frac{Sum\_giniSplit(\text{入学英语成绩}_F T)}{s} = \frac{\frac{5}{8}+\frac{3}{7}+\frac{17}{28}}{3} = \frac{31}{56}$$

根据改进后的公式计算出最后的信息增益率

$$GainR(\text{入学英语成绩}) = \frac{Gain(\text{入学英语成绩})\times M}{SplitInfo(\text{入学英语成绩}\ T)-\overline{Sum\_giniSplit(\text{入学英语成绩}_F T)}} = \frac{\sum_{i=1}^{n}\frac{|TC_i|\times(|T|-|TC_i|)}{|T|}-\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{|TC_{ij}|\times(|T_j|-|TC_{ij}|)}{|T_j|}}{\sum_{j=1}^{m}\frac{|T_j|\times(|T|-|T_j|)}{|T|}-\overline{Sum\_giniSplit(\text{入学英语成绩}_F T)}}\times M$$

其中

$$\sum_{i=1}^{n}\frac{|TC_i|\times(|T|-|TC_i|)}{|T|} = \frac{8\times(14-8)}{14}+\frac{6\times(14-6)}{14} = \frac{48}{7}$$

$$\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{|TC_{ij}|\times(|T_j|-|TC_{ij}|)}{|T_j|} = \frac{2\times(5-2)}{5}+\frac{3\times(5-3)}{5}+\frac{1\times(3-1)}{3}+\frac{2\times(3-2)}{3}+\frac{4\times(6-4)}{6}+\frac{2\times(6-2)}{6} = \frac{32}{5}$$

$$\sum_{j=1}^{m}\frac{|T_j|\times(|T|-|T_j|)}{|T|} = \frac{5\times(14-5)}{14}+\frac{3\times(14-3)}{14}+\frac{6\times(14-6)}{14} = 9$$

则“入学英语测试成绩”的信息增益率为

$$GainR(\text{入学英语成绩}) = \frac{\frac{48}{7} - \frac{32}{5}}{9 - \frac{31}{56}} \times 3 = 0.162367864$$

同理，可以计算出其它几个属性的信息增益率为

$$GainR(\text{性别}) = 0.059303187$$

$$GainR(\text{平时成绩}) = 0.192091941$$

$$GainR(\text{学习情况}) = 0.467447264$$

根据改进后的计算结果可得到

$$GainR(\text{学习情况}) > GainR(\text{平时成绩}) > GainR(\text{入学英语成绩}) > GainR(\text{性别})$$

这跟传统的 C4.5 算法计算结果有点一样，运用 G_C4.5 算法计算出来的结果应该选择“学习情况”为根属性，而“性别”属性是对“英语统考成绩”是否合格影响最小的因素，经过仔细分析可知，G_C4.5 算法的计算结果更具有准确性，即改进后的算法对于属性的选择更加准确。所以 G_C4.5 算法对于消除非类属性间的冗余度的影响的改进是成功的。

3.2 实验二

为验证改进后算法的性能的优越性，在怀卡托智能分析环境[12]（Waikato environment for knowledge analysis）下进行实验，WEKA 的版本是 3.7.11，改进后的算法跟原来算法进行性能对比实验，即还需要添加自己的算法，因此还配置了 Java 的 JDK、JRE 和 Eclipse，在 Eclipse 中重新编译运行 WEKA，这里所用的 JDK 版本为 jdk1.7.0_51，Eclipse 的版本是 4.4.0 版。

实验选用 UCI 数据集中的 sonar，anneal 和 sick 数据集进行对比实验，这 3 个数据集详细信息见表 2。

表 2 实验数据集简介

数据集	数据大小	属性个数	属性类型
Sonar	208	61	数值型
Anneal	898	39	离散型
Sick	3772	30	离散型

进入 WEKA 的实验界面，添加数据集和算法，设置实验迭代次数为 10。通过 10 次迭代实验的时间来计算出平均一次的执行时间（精确到秒），以及平均的分类准确率。

（1）用改进后的 G4.5 算法与传统的 C4.5 算法（改进前）进行比较，得到结果见表 3。

表 3 算法性能对比实验结果 1

数据集	算法	建模速度	分类准确率/%
sonar	C4.5	0.3 s	98.44
	G_C4.5	0.2 s	99.82
anneal	C4.5	0.7 s	73.61
	G_C4.5	0.5 s	92.78
sick	C4.5	0.9 s	98.82
	G_C4.5	0.6 s	99.69

其中，表中的执行时间为大约的执行时间，因为 WEKA 实验界面中的 LOG 只显示到“秒”。根据表 3 的结果可知，G_C4.5 算法的性能优于 C4.5 算法：不仅提高了运算效率也提高了分类准确率，由此可见，算法的改进是成功的。

（2）用改进后的 G4.5 算法与文献[5]中改进的算法（记为 C4.5_3 算法）进行比较，得到结果见表 4。

表 4 算法性能对比实验结果 2

数据集	算法	建模速度	分类准确率/%
sonar	C4.5_3	0.2 s	99.42
	G_C4.5	0.2 s	99.82
anneal	C4.5_3	0.5 s	90.72
	G_C4.5	0.5 s	92.78
sick	C4.5_3	0.5 s	99.11
	G_C4.5	0.6 s	99.62

根据表 4 的结果可以得出，改进后的算法 G_C4.5 相较于文献[5]中的算法（C4.5_3）来说，虽然在建模时间上略逊于 C4.5_3 算法，但是差值不大；而从分类准确度来说，C4.5_3 算法具有更高的分类准确率。因而总体上来讲，G_C4.5 算法更具有优越性。

4 结束语

运用数学上的泰勒级数和等价无穷小的原理，简化传统的 C4.5 算法中信息增益率的计算过程，并用属性的属性值个数来减少化简时所带来的误差。然后再引入非类属性之间的 gini 指数的平均值来消除非类属性间关联性对属性选择的影响，得到新的信息增益率的计算方法。经实验结果验证，改进后的算法（G_C4.5）计算效率更高，分类更加合理准确。实现了改进的目的。然而在实际应用中，特别是面对大数据挑战，如何使算法更加高效地处理海量的大数据，并得到更加准确的价值信息，是今后的研究方向。

参考文献：

[1] WANG Miao, CHAI Ruimin. Improved classification attribute selection scheme for decision tree [J]. Computer Engineering and Applications, 2014, 46 (8): 127-129 (in Chinese). [王苗，柴瑞敏. 一种改进的决策树分类属性选择方法 [J]. 计算机工程与应用，2014，46 (8)：127-129.]

[2] Han J, Mickeline K, Pei J. Data mining: Concepts and techniques [M]. FAN Ming, MENG Xiaofeng, transl. Beijing: China Machine Press, 2013 (in Chinese). [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，2013.]

[3] NI Zhiwei. Business intelligence and data mining [M]. Beijing: Beijing University Press, 2013 (in Chinese). [倪志伟. 商务智能与数据挖掘 [M]. 北京：北京大学出版社，2013.]

（下转第 1361 页）

有进行相应的算法复杂度的分析，需要在以后的研究加以考虑，以提高该方法的适用性。

参考文献：

[1] Tang Deyan，Zhu Ningbo，Yu Fu，et al. A novel sparse representation method based on virtual samples for face recognition [J]. Neural Computing & Applications，2014，24 (3)：513-519.

[2] Khashman Adnan. Application of an emotional neural network to facial recognition [J]. Neural Computing & Applications，2009，18 (4)：309-320.

[3] Zhang Baocheng，Yu Qiao. Face recognition based on gradient Gabor feature and efficient kernel fisher analysis [J]. Neural Computing & Applications，2010，19 (4)：617-623.

[4] Zhao Guoying，Huang Xiaohua，Taini Matti，et al. Facial expression recognition from near-infrared videos [J]. Image and Vision Computing，2011，29 (9)：607-619.

[5] Xu Yong，Zuo Wangmeng，Fan Zizhu. Supervised sparse representation method with a heuristic strategy and face recognition experiments [J]. Neurocomputing，2012，79 (1)：125-131.

[6] Zhang Lei，Yang Meng，Feng Xiangchu. Sparse representation or collaborative representation：Which helps face recognition？[C] //Proc of the IEEE International Conference on Computer Vision. Barcelona：IEEE，2011：471-478.

[7] Mehrdad J Gangeh，Ali Ghodsi，Mohamed S Kamel. Kerne lized supervised dictionary learning [J]. IEEE Transactions on Signal Processing，2013，61 (19)：4753-4767.

[8] Lai Zhihui，Jin Zhong，Yang Jian，et al. Sparse local discriminant projections for feature extraction [C] //Proceedings of ICPR，2010：926-929.

[9] Zhu Ningbo，Li Shengtao. A Kernel-based sparse representation method for face recognition [J]. Neural Computing & Applications，2014，24 (3-4)：845-852.

[10] YU Xu，YANG Jing，XIE Zhiqiang. Research on virtual sample generation technology [J]. Computer Science，2011，38 (3)：16-19 (in Chinese). [于旭，杨静，谢志强. 虚拟样本生成技术研究 [J]. 计算机科学，2011，38 (3)：16-19].

[11] Xu Yong，Zhu Qi. A simple and fast representation-based face recognition method [J]. Neural Computing & Applications，2013，22 (7-8)：1543-1549.

(上接第 1270 页)

[4] Kalpesh Adhatrao，Aditya Gaykar，Amiraj Dhawan，et al. Predicting students' performance using ID3 and C4.5 classification algorithms [J]. International Journal of Data Mining & Knowledge Management Process，2013，3 (5)：39-53.

[5] CHEN Ying，MA Zhongbing，HUANG Min. Improved algorithm of C4.5 decision tree [J]. Software，2013，34 (2)：61-64 (in Chinese). [陈英，马仲兵，黄敏. 优化的 C4.5 决策树算法 [J]. 软件，2013，34 (2)：61-64.]

[6] James S Sochacki. The modified picard method for solving arbitrary ordinary and initial value partial differential equations [D]. James Madison University，2012.

[7] ZHAO Weidong. Business intelligence [M]. Beijing：Tsinghua University Press，2012 (in Chinese). [赵卫东. 商务智能 [M]. 北京：清华大学出版社，2012.]

[8] WEI Hao，DING Yaojun. An improved algorithm of C4.5 decision tree based on attributes correlation [J]. Journal of North University of China (Natural Science Edition)，2014，35 (4)：2-6 (in Chinese). [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)：2-6.]

[9] HUANG Aihui. C4.5 algorithm of decision tree improvement and application [J]. Science Technology and Engineering，2009，9 (1)：34-36 (in Chinese). [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] LI Rui，CHENG Yanan. An improved C4.5 algorithm [J]. Science Technology and Engineering，2010，10 (27)：70-74 (in Chinese). [李瑞，程亚楠. 一种改进的 C4.5 算法 [J]. 科学技术与工程，2010，10 (27)：70-74.]

[11] LIU Xiaoyu，WEI Zhiqiang. An improved C4.5 algorithm and application [D]. Qingdao：Institute of Information Science and Engineering of Ocean University of China，2013 (in Chinese). [刘晓宇，魏志强. C4.5 算法的一种改进及其应用 [D]. 青岛：中国海洋大学信息科学与工程学院，2013.]

[12] Machine learning group at the university of Waikato [EB/OL]. http://www.cs.waikato.ac.nz/ml/.