计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(7):139−144 [doi: 10.15888/j.cnki.csa.006420]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于粗糙集理论与 CAIM 准则的 C4.5 改进算法①

于宏涛, 贾宇波

(浙江理工大学 信息学院, 杭州 310018)
通讯作者: 贾宇波, E-mail: jiayubo1964@163.com

摘　要: C4.5 算法是一种非常有影响力的决策树生成算法, 但该方法生成的决策树分类精度不高, 分支较多, 规模较大. 针对 C4.5 算法存在的上述问题, 本文提出了一种基于粗糙集理论与 CAIM 准则的 C4.5 改进算法. 该算法采用基于 CAIM 准则的离散化方法对连续属性进行处理, 使离散化过程中的信息丢失程度降低, 提高分类精度. 对离散化后的样本用基于粗糙集理论的属性约简方法进行属性约简, 剔除冗余属性, 减小生成的决策树规模. 通过实验验证, 该算法可以有效提高 C4.5 算法生成的决策树分类精度, 降低决策树的规模.

关键词: 粗糙集; 离散化; 属性约简; 决策树; CAIM; C4.5

引用格式: 于宏涛,贾宇波.基于粗糙集理论与 CAIM 准则的 C4.5 改进算法.计算机系统应用,2018,27(7):139–144. http://www.c-s-a.org.cn/1003-3254/6420.html

C4.5 Improved Algorithm Based on Rough Set Theory and CAIM Criterion

YU Hong-Tao, JIA Yu-Bo

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: As a decision tree generated algorithm, C4.5 algorithm is very influential. But the decision tree classification by C4.5 algorithm is of less accuracy, more branches, and larger scale. To solve these problems, we propose a C4.5 improved algorithm based on rough set theory and CAIM criterion. The algorithm uses the discretization method based on CAIM criterion to process the continuous attributes, which decreases the information loss degree and improve the classification accuracy in discretization. The discretized sample is reduced by attribute reduction method based on rough set theory, which eliminates the redundant attribute and trims the size of decision tree. Experiments show that the algorithm can effectively improve the classification accuracy of decision tree generated by C4.5 algorithm and reduce the scale of decision tree.

Key words: rough set; discretization; attribute reduction; decision tree; CAIM; C4.5

决策树算法在数据挖掘领域占有非常重要的地位. 当前主要的决策树算法[1]包括 ID3 算法、C4.5 算法、CART 算法、SLIQ 算法、SPRINT 算法、PUBLIC 算法等. C4.5 算法[2]是 1993 年由 Quinlan 提出的, 其因直观高效等优点在医疗、金融、教育、互联网等多个领域得到广泛应用. 徐鹏等[3]利用 C4.5 算法对网络流量进行了分类; 罗森林等[4]将 C4.5 算法引入到糖尿病的数据处理中, 建立了有效的预测糖尿病的规则; 周琦[5]采用 C4.5 算法对学生成绩进行分析, 对学生高考成绩进行了预测; 吕晓丹[6]利用 C4.5 算法为商业银行建立了企业信用评价模型. 该算法的基本思想是首先将整个数据集做为根节点, 用各个属性的信息增益率作为度量属性重要程度的标准, 选择信息增益率最大的属性作为分裂属性将根节点划分成若干个子节点,

① 收稿时间: 2017-11-04; 修改时间: 2017-11-27; 采用时间: 2017-12-01; csa 在线出版时间: 2018-06-27

并在各个子节点上重复进行分裂操作生成一棵完整的决策树.

C4.5 算法克服了 ID3 算法偏向于选择属性值较多的属性的缺点, 新增了对连续属性的处理方法. 但是该算法生成的决策树分类精度不高, 树的规模较大. 造成上述问题的原因为: 1) C4.5 算法在选择连续属性的分裂点时只考虑单个属性与类别属性的关系, 割裂了各个属性之间的联系, 这就造成了信息损失, 导致最终的决策树分类精度下降. 2) 在一个信息系统中, 并不是所有属性都是同等重要的, 有些属性与分类无关, 这些冗余属性的存在, 使生成的决策树规模较大, 分支增多. 当前有许多学者对 C4.5 算法进行了优化, 刘佳等[7]基于 Fayyad 和 Irani 的证明, 对 C4.5 算法的连续属性离散化等方面进行了改进, 提高了算法效率和分类准确率; 曹燕等[8]提出了一种粗糙集理论与 C4.5 算法相结合的算法 RSC4.5, 降低了分类结果的复杂度; 黄秀霞等[9]将泰勒公式引入到信息增益的计算公式内, 提高了决策树的构建效率.

本文提出了一种基于粗糙集理论与 CAIM 准则的 C4.5 改进算法. 本算法在连续属性的处理上采用了一种基于 CAIM 准则的多属性关联的离散化算法, 使得在连续属性离散化过程中造成的信息损失降低, 提高了决策树的分类精度; 运用一种基于粗糙集理论的属性约简算法去除了决策表中与分类无关的条件属性, 使得生成决策树的规模减小.

1 基于 CAIM 准则的离散化方法

C4.5 算法的一个优点是支持对连续型数据的处理, 其处理方式为对首先每个连续属性选取足够多的候选断点, 再分别计算每个候选断点的信息增益, 选取信息增益最大的断点作为最终断点. 但是 C4.5 算法的断点选择过程中只考虑了单个连续属性与决策属性之间的关系, 这中处理方式造成了较多的信息损失, 使得构建的决策树分类精度降低. 杨萍等[10]提出了一种基于 CAIM 准则的数据离散化方法, 该算法根据决策表中决策属性与多个条件属性之间的联系构造离散化框架, 使得离散化后的条件属性与决策属性的关联程度达到最大. 利用这种离散化方法可以降低连续属性离散化过程中的信息损失, 有效提高 C4.5 算法的分类精度.

1.1 CAIM 准则

Kurgan 和 Cios 在 2004 年提出了 CAIM 准则[11], 它是类别和属性值之间的关联程度的一种度量. 下面对 CAIM 准则进行简单的介绍.

给定一个决策表 S, 条件属性集为 C, C 中包含 w 个连续属性, 决策属性为$\{d\}$. 决策表 S 共包含 n 个元素, 决策属性 d 将所有元素划分为 k 个类别. 对 C 中任意一个连续属性 c, 可以确定一个离散化框架, 将 c 的值域划分为 m 个离散区间 $\{[c_0,c_1],(c_1,c_2],\cdots,(c_{m-1},c_m]\}$, c_0 为属性 c 值域中的最小值, c_m 为最大值, 对于任意的$0 \le i \le m$, 有 $c_i>c_{i-1}$.

根据一个确定的离散化框架、决策属性变量和连续属性 c 的离散区间, 可以确定一个二维矩阵, 如表 1 所示. 其中$\{d_1 \cdots d_i \cdots d_k\}$表示决策属性 d 的 k 个取值, e_{ir} 表示在决策表中决策属性取值为 d_i 且连续属性 c 的取值在 $(c_{r-1}, c_r]$中的元素个数. N_{di} 表示决策属性取值为 d_i 的元素个数, N_{cr} 表示属性 c 的取值在区间 $(c_{r-1}, c_r]$的元素个数.

表 1 离散化二维矩阵

类别/区间	$[c_0, c_1]$	...	$(c_{r-1}, c_r]$	...	$(c_{m-1}, c_m]$	总和
d_1	e_{11}	...	e_{1r}	...	e_{1m}	N_{d1}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
d_i	e_{i1}	...	e_{ir}	...	e_{im}	N_{di}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
d_k	e_{k1}	...	e_{kr}	...	e_{km}	N_{dk}
总和	N_{c1}	...	N_{cr}	...	N_{cm}	N

由一个离散化二维矩阵可得 CAIM 准则的如下定义:

$$CAIM = \frac{1}{m}\sum_{r=1}^{m}\frac{\{e_{ir}\}_{\max}^{2}}{N_{cr}}$$

$\{e_{ir}\}_{\max}$ 表示当 i 变化时 e_{ir} 的最大值, 如果$\{e_{ir}\}_{\max} = e_{hr}$, $(1 \le h \le k)$, 则称决策属性取值为 d_h 的元素构成的集合为区间 $(c_{r-1}, c_r]$中的主导类. 主导类中包含的元素个数越多, $CAIM$ 的值就越大, 类别与离散区间的关联程度就越大.

1.2 一种基于 CAIM 准则的离散化算法

离散化算法的根本目的是求取一个断点集合, 将连续属性划分为多个区间. 在最理想的情况下, 可以选取出 k–1 个断点 (k 为决策属性取值个数), 断点划分出的 k 个区间中的元素全部属于同一个类别, 此时 $CAIM$ 达到理论上的最大值 $CAIM=n/k$, k–1 个断点为最佳划分断点集合. 在实际问题中 $CAIM$ 的值随着断点的增加而增加, 达到局部最大后开始下降, 此时为使

决策属性与待离散的条件属性关联程度达到最大, 必须要选择足够多的断点, 使划分出的离散区间中所有元素都属于同一个类别. 如果只考虑离散区间中的主导类则会造成信息损失, 使离散结果的质量下降.

前面论述的都是单个连续条件属性与决策属性之间的关联, 实际上仅仅依靠单个属性是不能区分元素的, 因此应该考虑到决策属性与所有属性之间的关联关系, 从而降低信息损失, 提高分类的精度. 从多个属性的角度考虑连续属性的离散化, 实质上就是寻找一个最小的断点集合, 这些断点可以将整个决策表划分成一个个的超立方体, 并使得每个超立方体中的元素都属于同一个类别. 在考虑多个属性与决策属性关联的离散化算法时如果只追求由断点集划分出的超立方体空间中的元素都属于同一个类别, 则会使断点个数太多, 离散效果不好. 因此我们需要在断点个数与超立方体空间中元素纯度之间寻找一个平衡, 使离散化结果达到我们想要的效果. 由前面的讨论我们知道在最理想的情况下每个连续属性的断点个数为 k–1 个, 将每个连续属性划分为 k 个离散区间, 在此我们将 k 设为离散化区间的期望个数. 在每个离散化区间中, 设定一个最小阈值, 使得每个区间中主导类的个数不小于最小阈值, 给出一个常量 R, $0.5 \le R \le 1$, 则有 $\{e_{ir}\}_{\max} > R*N_{cr}$, $r=1, 2, \cdots, k$. 此时有最小的 $CAIM$ 期望值为:

$$CAIM_{\exp} = \frac{1}{k}\sum_{r=1}^{m}\frac{(R*N_{cr})^2}{N_{cr}} = \frac{N*R^2}{k}$$

设 $CAIM_{\text{pmax}}$ 为局部最大值, 则每增加一个断点 $CAIM$ 的下降幅度不超过 $(CAIM_{\text{pmax}}-CAIM_{\exp})/(k-1)$ 时, 可以得到预期的离散化结果.

给定一个决策表 $S=(U, C\cup D, V, f)$, 其中 U 为论域, C 为条件属性集合, D 为决策属性集合, V 为值域, f 为属性与值域间的函数关系, 对于有多个决策属性的决策表, 可以转化为单一决策属性的决策表. 设决策表包含 n 个元素即 $U=\{x_1, x_2, \cdots, x_n\}$, 连续属性集合 C 中包含 l 个连续属性 $\{c_1, c_2, \cdots, c_l\}$, 决策属性 d 共有 k 个取值 $\{d_1, d_2, \cdots, d_k\}$ 可以将决策表划分为 k 个类别的小决策表 $(U_1, U_2, \cdots, U_k)$. V_{ci} 为属性 c_i 的值域. 定义常量 R, 且 $0.5 \le R \le 1$. 具体离散化算法步骤如算法 1.

算法 1. 基于 CAIM 准则的多属性关联离散化算法

1) 设 $i=1$, DS=\{U\}, 离散化结果断点集合 CutPoint=Φ.

2) 设 $DS_{temp}=\Phi$, $r=1$. 对 DS 中的一个元素 U_r, U_r 内共包含 k_r 个类别.

3) 取一个连续属性 c_i, 将 c_i 的值域按照升序排列得: $V_{ci}^r=\{v_{ci}^0, v_{ci}^1, \cdots, v_{ci}^m\}$, 取两个相邻属性值的平均数构成集合 $B=\{(v_{ci}^0+v_{ci}^1)/2, (v_{ci}^1+v_{ci}^2)/2, \cdots, (v_{ci}^{m-1}+v_{ci}^m)/2\}$. B 为候选断点集. 初始化离散化框架为 $LS=\{[v_{ci}^0, v_{ci}^m]\}$, 历史 $CAIM$ 最大值 $CAIM_{\text{pmax}}=0$, 属性 c_i 的断点集合 Point=Φ.

4) 设 $j=1$, $CAIM$ 的最小期望值为 $CAIM_{\exp}=|U^r|R^2/k^r$.

5) 遍历 B 中的断点加入离散化框架 LS 中计算 $CAIM$ 值, 取使 $CAIM$ 值最大的点, 记 $CAIM$ 最大值为 $CAIM_h$.

6) 如果第 5 步中 $CAIM_h$ 满足 $CAIM_h > CAIM_{\text{pmax}}$ 或 $(CAIM_{\text{pmax}}-CAIM_h) < (CAIM_h-CAIM_{\exp})$ (k_r-1), 则令 $CAIM_{\text{pmax}}=CAIM_h$. 将第 4 步选择的断点加入离散化框架 LS 和断点集合 Point 中, 并在 B 中删除该断点. 另 $j=j+1$, 如果 $j<k^r$ 则转到第 5 步.

7) 假设离散化框架 LS 将 U_r 划分成多个小集合, 设 U_x 是其中一个, 如果 U_x 中的元素不都属于同一类别则另 $DS_{temp}=DS_{temp}\cup\{U_x\}$.

8) 令 $DS=DS_{temp}$, $r=r+1$, 如果 $r\le|DS|$, 则转至第 3 步.

9) 此时 Point 为属性 c_i 的断点集合, 令 CutPoint=CutPoint∪Point, $i=i+1$, 如果 $i<l$, 转至第 2 步, 否则算法结束, CutPoint 为各个连续属性的断点集合.

2 基于粗糙集理论的属性约简

2.1 粗糙集理论

1982 年 Pawlak 提出了粗糙集理论[12], 它是一种新的处理不确定、不精确和含糊信息的数据分析理论. 下面对本文用到的粗糙集理论相关概念进行简单介绍.

定义 1. 给定一个决策表 $S=(U, C\cup D, V, f)$, 其中 $U=\{x_1, x_2, \cdots, x_n\}$ 是决策表中所有对象的集合, 称为论域; $C\cup D$ 为决策表中全部属性, 其中 C 代表条件属性, D 代表决策属性; V 是属性值域, f 是一个信息函数, 表示了属性和元素的对应关系. 设 P 是一个属性子集, 且 P 不为空, 则 P 决定了一个不可分辨关系 IND(P). IND(P) 对论域 U 构成了一个划分, 用 U/IND(P) 表示. 设 $X\subseteq U$ 和一个属性子集 G, 定义子集 X 关于 G 的下近似和上近似分别为:

$$\underline{G}(X) = \cup\{Y|(\forall Y\in U/IND(G))\wedge(Y\subseteq X)\}$$

$$\overline{G}(X) = \cup\{Y|(Y\in U/IND(G))\wedge(Y\cap X\ne\Phi)\}$$

定义集合 $bng(X)=\overline{G}(X)-\underline{G}(X)$ 为 X 的 G 边界域; 集合 $pos_G(X)=\underline{G}(X)$ 为 X 的 G 正域; 集合 $neg_G(X)=U-\overline{G}(X)$ 为 X 的 G 负域.

定义 2. 设 P 和 Q 分别是决策表 S 的属性子集, 则 P, Q 分别将论域 U 构成一个划分 X=U/IND(P) 和 Y=U/IND(Q) 记 $X=\{X_1, X_2, \cdots, X_n\}$, $Y=\{Y_1, Y_2, \cdots, Y_m\}$, 将 X 和 Y 在论域 U 上的概率分布表示为:

$$p(X_i)=\frac{card(X_i)}{card(U)},\quad p(Y_i)=\frac{card(Y_i)}{card(U)}$$

其中 $card$(E) 表示集合 E 中的元素个数. 记 $p(Y_j|X_i)$ 为 Y_j 对 X_i 的条件概率, 计算公式如下:

$$p(Y_j|X_i)=\frac{card(X_i\cap Y_j)}{card(X_i)}$$

定义 3. 对于给定的属性子集 P 和他的概率分布, 称 H(P) 为 P 的信息熵, 计算公式如下:

$$\mathrm{H(P)}=-\sum_{i=1}^{n}p(X_i)\log p(X_i)$$

定义 4. 对于给定的属性子集 P 和 Q 以及它们的概率分布和条件概率分布, 称 H(Q|P) 为 Q 相对于 P 的条件熵, 计算公式如下:

$$H(Q|P)=-\sum_{i=1}^{n}p(X_i)\sum_{j=1}^{m}p(Y_j|X_i)\log p(Y_j|X_i)$$

定义 5: 给定属性子集 P、Q 以及他们的信息熵和条件熵, 称 H(Q)-H(Q|P) 为 P 与 Q 的互信息, 记为 I(P;Q).

定义 6[13]. 对于决策表 S, 条件属性集为 C, 决策属性集为 D, 取 $p\in$C, 如果条件熵 H(D|C)=H(D|(C-{p})), 则称属性 p 在 C 中是 D-不必要的; 如果条件熵 H(D|C)<H(D|(C-{p})), 则称属性 p 在 C 中是 D-必要的. 如果 C 中所有属性都是 D-必要的, 则称 C 是相对于决策 D 独立的. C 中所有 D-必要的属性组成的集合称为 C 的相对于决策 D 的核, 记为 $\mathrm{CORE_D(C)}$.

定义 7. 对于决策表 S, 条件属性集为 C, 决策属性集为 D, 设B $\subseteq$ C, 如果满足I(B;D) = I(C;D)且$\forall p\in B$, 都有H(D|B) < H(D|(B − {p}))则称 B 是 C 的一个相对决策 D-约简, 记为B $\in \mathrm{RED_C(D)}$, $\mathrm{RED_C(D)}$是所有约简组成的集合.

2.2 属性约简算法

属性约简是运用粗糙集理论处理的主要问题之一. 属性约简的根本目的是在保证决策表分类能力不变的前提下删除条件属性中的冗余属性. 当前有很多有效的属性约简算法, 如基于属性重要度的约简算法[14]、基于信息熵的约简算法[15]、基于互信息的属性约简算法等[16]. 本文将采用基于互信息的属性约简算法对决策表进行属性约简, 其约简算法步骤如算法 2.

算法 2. 基于互信息的属性约简算法

1) 计算决策表中条件属性 C 和决策属性 D 的互信息 I(C;D)=H(D)−H(D|C).
2) 计算 C 相对于 D 的核属性集合 $\mathrm{CORE_D(C)}$, 另 B=$\mathrm{CORE_D(C)}$.
3) 计算 B 与决策属性 D 的互信息 I(B;D)=H(D)−H(D|B), 如果 I(B;D)=I(C|D) 则转到第 5 步.
4)取任意的条件属性 c_i,$c_i\in$(C−B),计算互信息I(c_i,D|B)=H(D|B)−H(D|B∪{c_i}), 求得使 I(c_i, D|B) 最大的属性 c_m, 令B=B∪c_m, 转到第 3 步.
5) 输出 B, B 即为决策表的属性约简.

3 基于粗糙集理论与 CAIM 的 C4.5 算法

3.1 C4.5 算法

C4.5 算法是一种在 ID3 算法基础上改进的决策树生成算法, 它采用信息增益率作为节点选择衡量标准, 在 ID3 算法的基础上增加了对连续属性的处理和剪枝方法. C4.5 算法整体上分为 3 步: 1) 如果决策表中有连续属性, 对连续属性进行离散化处理. 2) 以信息增益率做为分裂节点的选择标准, 递归构建决策树. 3) 对构建的决策树采用悲观剪枝方法[17]进行剪枝处理. 下面首先介绍信息增益率的相关概念, 再对 C4.5 算法的步骤进行简单说明.

定义 8. 给定一个决策表 S, 决策属性将整个决策表划分为 m 个样本子集S = {$S_1,S_2,\cdots,S_m$}. p_i 表示决策表中一个元素分布在 S_i 中的概率, S 的信息熵定义为:

$$Entropy(\mathrm{S})=-\sum_{i=1}^{m}p_i\log p_i$$

定义 2. 给定决策表 S, 条件属性 c 将决策表划分为 k 个不同样本子集S = {$S_1,S_2,\cdots,S_k$}, 则按属性 c 划分 S 的信息增益定义如下:

$$Gain(\mathrm{S},c)=Entropy(\mathrm{S})-Entropy_c(\mathrm{S})$$

$$Entropy_c(\mathrm{S})=-\sum_{i=1}^{k}\frac{card(S_i)}{card(\mathrm{S})}Entropy(S_i)$$

定义 3. 信息增益率定义如下:

$$GainRatio(\mathrm{S},c)=\frac{Gain(\mathrm{S},c)}{SplitE(\mathrm{S},c)}$$

$$SplitE(\mathrm{S},c)=-\sum_{i=1}^{k}\frac{card(S_i)}{card(\mathrm{S})}\log\frac{card(S_i)}{card(\mathrm{S})}$$

C4.5 算法的具体步骤如算法 3.

算法 3. C4.5 算法

1) 如果决策表中有连续属性, 则首先对决策表的连续属性变量进行离散化处理. 设连续属性有 n 个取值, 处理过程为首先对连续属性取值进行从小到大排序, 再选择相邻值的平均值作为分割点将连续属性划分为 n 个小区间.

2) 计算每个属性的信息增益率. 对于连续属性, 分别计算以候选分割点进行划分的信息增益率, 选择信息增益率最大的候选分割点作为该连续属性的最终分割点. 比较各个属性的信息增益率, 选择信息增益率最大的属性作为分裂节点.
3) 分裂节点的每个属性取值都对应一个子集, 再对子集递归执行第 2 步, 直到划分的每个子集中的元素都属于同一类别, 生成决策树.
4) 对第 3 步生成的决策树采用悲观剪枝方法进行剪枝处理.

3.2 改进的 C4.5 算法

通过对 4.5 算法的分析, 发现以下问题导致 C4.5 算法生成的决策树分类精度不高, 树的规模较大. 1) C4.5 算法对连续属性的处理只考虑了单一连续属性与决策属性的关联关系, 在离散化过程中会造成较多的信息损失, 导致分类精度降低. 2) 决策表中可能存在与分类无关的冗余属性, 使得决策树构建过程中生成树的规模过大.

为解决上述问题, 本文提出了一种基于 CAIM 准则和粗糙集理论的 C4.5 优化算法. 其基本思想是首先判断决策表中是否有连续属性, 如果有则采用基于 CAIM 准则的数据离散化方法对连续属性进行离散化处理. 再利用基于粗糙集理论的属性约简算法对决策表进行属性约简, 最后计算每个属性的信息增益率, 选择分裂属性递归构造决策树. 此算法采用的离散化方法考虑了多个条件属性与决策属性的关联关系, 避免了 C4.5 算法在连续属性的处理过程中只考虑单一属性与决策属性关系所造成的信息损失, 可有效提高决策树的分类精度. 改进的 C4.5 算法在构建决策树之前增加了属性约简的步骤, 属性约简可以剔除与分类无关的条件属性, 使生成的决策树规模减小. 该算法具体步骤如算法 4.

算法 4. 基于 CAIM 准则与粗糙集理论的 C4.5 改进算法

1) 如果决策表中有连续属性, 运用于 CAIM 准则的离散化方法对连续属性进行离散化处理, 否则转至第 2 步.
2) 运用基于粗糙集理论的属性约简算法对决策表进行属性约简, 去除冗余属性.
3) 计算每个属性的信息增益率, 选择信息增益率最大的属性作为分裂节点.
4) 分裂节点的属性取值将决策表划分为多个小决策表, 对每个小决策表递归执行第 3 步, 直到划分出的每个小决策表中的元素都属于同一类别, 生成决策树.
5) 对第 4 步生成的决策树采用悲观剪枝方法进行剪枝处理.

4 实验设计与分析

为验证本文改进算法的有效性, 从 UCI 机器学习数据库[18]中选取了 3 个有代表性的决策数据集: 虹膜植物集 iris、葡萄酒识别数据集 wine、皮马印第安人糖尿病数据集 pima 进行实验分析. 这 3 个数据集的详细信息如表 2 所示.

表 2 实验数据

数据集	实例个数	条件属性个数	连续条件属性个数	决策属性个数	类别个数
iris	150	4	4	1	3
pima	768	8	8	1	2
wine	178	13	13	1	3

取每个数据集的 70% 作为训练样本数据, 30% 为测试数据. 将本文算法中离散化过程的参数 R 设为 0.8 与 Weka 中的 J48(即 C4.5) 算法在分类准确率与生成决策树节点个数两个方面进行对比, 实验结果如表 3 所示.

表 3 实验结果

数据集	算法	准确率 (%)	树节点个数
iris	C4.5	96	9
	本文算法	98.13	6
pima	C4.5	73.82	39
	本文算法	79.69	28
wine	C4.5	93.82	9
	本文算法	96.13	7

从表 3 可以看出, 本文算法与 C4.5 算法相比 iris 数据集准确率提高了 2.13%, 树节点个数减少 3 个; pima 数据集准确率提高了 5.78%, 树节点个数减少 8 个; wine 数据集准确率提高了 2.31%, 树节点个数减少了 2 个. 从实验结果中可以看出, 本文算法相较于 C4.5 算法在分类准确率和生成树规模上都有一定程度的优化, 证明本文算法切实有效可行. 通过本文算法生成的决策树能够更好的对新数据进行分类和预测.

5 结论

本文对基于 CAIM 准则的离散化方法、基于粗糙集理论的属性约简算法和 C4.5 算法进行了简单介绍. 通过对 C4.5 算法分类精度不高、生成决策树规模较大的问题产生原因进行分析. 本文用基于 CAIM 准则的离散化方法代替 C4.5 算法中对连续属性的处理方法, 在离散化的基础上进行属性约简, 去除冗余属性, 降低了决策表的属性维度, 最后生成决策树. 实验结果表明改进后的 C4.5 算法有更好的分类效果, 生成更为简洁的决策树.

参考文献

1 谢妞妞. 决策树算法综述. 软件导刊, 2015, 14(11): 63–65.

2 Quinlan JR. C4.5: Programs for machine learning. San Francisco: Morgan Kaufmann Publishers Inc., 1993.

3 徐鹏, 林森. 基于 C4.5 决策树的流量分类方法. 软件学报, 2009, 20(10): 2692–2704.

4 罗森林, 成华, 顾毓清, 等. C4.5 算法在 2 型糖尿病分类规则建立中的应用. 计算机应用研究, 2004, 21(7): 174–176, 179.

5 周琦. 改进的 C4.5 决策树算法研究及在高考成绩预测分析中的应用[硕士学位论文]. 南宁: 广西大学, 2012.

6 吕晓丹. 基于改进的决策树信用评价模型研究及其工具实现[硕士学位论文]. 上海: 东华大学, 2014.

7 刘佳, 王新伟. 一种改进的 C4.5 算法及实验分析. 计算机应用与软件, 2008, 25(12): 260–262. [doi: 10.3969/j.issn.1000-386X.2008.12.090]

8 曹艳, 殷旭. 一种基于粗糙集属性约简的 C4.5 算法. 北京信息科技大学学报, 2014, 29(6): 74–79.

9 黄秀霞, 孙力. C4.5 算法的优化. 计算机工程与设计, 2016, 37(5): 1265–1270, 1361.

10 杨萍, 杨天社, 杜小宁, 等. 一种基于类别属性关联程度最大化离散算法. 控制与决策, 2011, 26(4): 592–596.

11 Kurgan LA, Cios KJ. CAIM discretization algorithm. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(2): 145–153. [doi: 10.1109/TKDE.2004.1269594]

12 Pawlak Z. Rough sets. International Journal of Computer & Information Sciences, 1982, 11(5): 341–356.

13 苗夺谦, 王珏. 粗糙集理论中概念与运算的信息表示. 软件学报, 1999, 10(2): 113–116.

14 Hu XH. Knowledge discovery in databases: An attribute-oriented rough set approach[Ph. D Dissertation]. Regina: University of Regina, 1995.

15 王国胤, 于洪, 杨大春. 基于条件信息熵的决策表约简. 计算机学报, 2002, 25(7): 759–766.

16 苗夺谦, 胡桂荣. 知识约简的一种启发式算法. 计算机研究与发展, 1999, 36(6): 681–684.

17 黄文. 决策树的经典算法: ID3 与 C4.5. 四川文理学院学报 (自然科学), 2007, 17(5): 16–18.

18 Blake C. UCI repository of machine learning databases. ftp://ftp.ics.uci.edu/pub/machine-learning-databases, 1998.