计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(4):156−162 [doi: 10.15888/j.cnki.csa.007326]

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

基于粗糙集的决策树 ID3 算法①

余建军, 张琼之

(华南理工大学 工商管理学院, 广州 510640)
通讯作者: 张琼之, E-mail: 1010940616@qq.com

摘　要: 针对传统 ID3 算法计算过程复杂以及存在信息冗余的问题, 提出了一种改进算法——基于粗糙集属性约简的简化 ID3 算法. 该算法利用粗糙集中属性约简的性质删掉了系统中多余的知识, 在保证同样的分类能力下使得分类系统更简洁, 同时借助了泰勒公式对熵公式进行化简, 使得计算更简便, 然后把改进的算法用到实例中去, 并用相关数据库上的大量数据编程进行仿真实验, 最后得出的仿真结果证明了所提出算法的正确性与可行性, 不仅能够有效降低信息重复度,减少了冗余规则,还保证了算法精度, 同时为把 ID3 算法更好地应用到现实生活实例中提供了一定的参考价值.

关键词: 决策树; ID3 算法; 粗糙集; 属性约简; 仿真

引用格式: 余建军,张琼之.基于粗糙集的决策树 ID3 算法.计算机系统应用,2020,29(4):156–162. http://www.c-s-a.org.cn/1003-3254/7326.html

Decision Tree ID3 Algorithm Based on Rough Set

YU Jian-Jun, ZHANG Qiong-Zhi

(School of Business Administration, South China University of Technology, Guangzhou 510640, China)

Abstract: Aiming at solving the problem that the traditional ID3 algorithm is complicated and there exists redundancy information, this study proposes an improved algorithm—attribute reduct simplified ID3 algorithm based on rough set. This algorithm uses the properties of attribute reduct in rough set to delete the redundant knowledge, and makes the classification system more concise with the same classification ability. At the same time, it simplifies the entropy formula with Taylor formula to make the calculation easier. And then this study applies the improved algorithm to the example, and uses the massive data in the related database to program in order to do simulation experiments. Finally, the simulation results proved the correctness and feasibility of the proposed algorithm. It can not only reduce the information duplication, reduce the redundancy rules, but also ensure the accuracy of the algorithm. At the same time, it provides a certain reference value for the better application of ID3 algorithm to real life examples.

Key words: decision tree; ID3 algorithm; rough set; attribute reduct; simulation

引言

ID3 算法是著名的决策树, 它是以信息论为基础的一项从上而下的贪心算法, 基于熵的计算从根节点处的所有实例训练集合开始构造决策树. 它通过概率的相关运算来分析对象的属性值, 从而画出像一棵倒立的树形结构图来形象生动地解说实例, 以助于分析决策者做出预测决策. ID3 算法规则简单, 容易让人理解, 其算法与最基础的决策树算法[1]保持一致, 一般用来解决离散值一类的问题, 构造的树尤为形象地说明结果, 一目了然, 得到了众多学者的喜爱.

目前, ID3 算法是众多决策树构造算法中应用得最为广泛的一种, 如在医学、教学、公司业绩上的应用

① 基金项目: 广东省哲学社会科学“十三五”规划学科共建项目 (GD17XGL56)
Foundation item: Disciplinary Construction Project of Philosophic Social Science Fund of Thirteenth Five-Year Plan, Guangdong Province (GD17XGL56)
收稿时间: 2019-08-12; 修改时间: 2019-09-06; 采用时间: 2019-09-18; csa 在线出版时间: 2020-04-05

等. 孙道[2]把工厂所关注的特征变量对象返回的期望值传递给 ID3 算法中递归过程, 构造出决策树模型, 提高算法在其他应用中的移植性. 以上的学者是把决策树 ID3 算法运用到实际实例中去, 但并没有进行改进及优化算法, ID3 算法自身所带的局限性还是存在的, 还是有不少问题值得研究的. ID3 算法进行局部分析, 每次只选择一个属性分析, 从而产生大量规则以致效率会下降, 并且只能做到局部最优; 它优先考虑属性值数目最多的属性, 但问题是属性值多不一定是最优的测试属性; 它构造的决策树是单变量决策树, 而忽略了属性之间可能存在的相互依赖的关系; 它的计算方法比较复杂, 由于是关于对数 log 的运算, 所以运算量非常大; 它对信息系统的数据没有进行简化, 存在冗余的信息, 当系统大量信息时会变得复杂, 构造决策树所需的时间也会增多.

由于 ID3 算法只能处理离散属性值, Quinlan 又投入决策树算法的研究当中去, 于 1993 年提出了以 ID3 为基础核心的 C4.5 算法, 它不仅能处理离散属性值, 还能处理连续属性值, 并且继承了 ID3 算法的所有优点. 接着, Breman 等人提出了构造一颗二叉树的分类与回归树算——CART 算法, 它具有非常强大的统计解析能力, 处理数据后得到一颗二叉树, 简洁且易懂. 此外, IBM 的研究人员提出了 SLIQ 算法, 具有很好的伸缩性. 后来, 很多学者致力于研究多决策树综合技术, 探索出各种多决策树的分类方法, 例如 Schapire R 提出的 Boosting、Breiman 提出的 Bagging、Random Forest 分类方法等. 再后来, 决策树的增量算法出现了, 有增量 ID3、ID5R 等[3].

黄爱辉等人利用数学上等价无穷小的性质提出一种新的改进的 ID3 算法[4]. 杨霖等人提出了基于粗糙集、粒计算和分类矩阵的 ID3 改进算法[5]. 作者引入修正参数来改进 ID3 算法倾向于多值属性选取的问题[6,7]. 也有学者利用模糊规则提出了两种 ID3 改进算法[8,9]. 李建等人[10]在 ID3 的基础上引入属性重要度值用以计算新的信息熵, 并在信息增益计算中加入关联度函数比,提出了 AFIV-ID3 算法, 克服了 ID3 多值偏向的缺点. 大部分学者都是针对 ID3 取值较多的属性和只能处理离散型数据这两个缺陷进行改进及其优化算法的, 少有学者利用粗糙集中的属性约简去删除系统中多余的信息.

于天佑等人[11]研究表明在选定的特定类的数量相对全部决策类的数量较少时,约简的结果可能会更短, 约简的效率也会有不同程度的提升. 邬阳阳等人[12]当前研究中拓展粗糙集模型在约简理论完善、大数据处理、特殊数据处理等 3 个方面的问题依然存在. 尹继亮[13]利用区间值、正域, 并引入局部约简的概念, 设计了基于差别矩阵的局部约简算法. 粗糙集属性约简算法是一种数据预处理的有效方法,但指标多的时候, 使用区间值与正域求解是非常困难, 运行起来效率低下, 因此属性约简的应用研究相对较少. 少有学者应用分辨函数去求解属性约简, 更少学者想到要把 ID3 中计算熵公式中的复杂对数进行化简. 本文将介绍 ID3 的基本原理, 针对 ID3 信息系统存在冗余知识与计算方法复杂这两点缺点进行改进, 提出了基于粗糙集属性约简的简化 ID3 算法, 重点在于改进算法的实例分析、实验仿真和结果分析比较.

1 ID3 算法的基本理论

决策树算法这个说法最早出现于上世纪 60 年代, 到了 1979 年, 澳大利亚学者 Quinlan 提出了迭代两分器算法 (Iterative Dichotomizer3), 故简称 ID3 算法. 决策树是数据挖掘分类算法中的一个重要分支, 是一种归纳学习的贪心算法, 属于有监督学习法. 它主要是以实例为基础, 通过概率的相关运算来分析对象的属性值, 从而画出像一棵倒立的树形结构图来形象生动地解说实例, 以助于分析决策者做出预测决策. 下面简述一下 ID3 的基本理论.

对于一组实例训练样本集合 U, 共有 n 个样本集合; 分类属性集合为 C, 有 c 个不同的类; 决策属性集合为 D, 将实例训练集合分为 d 个不同的类$D_i(i=1,2,\cdots,d)$, 每个类的个数为$n_i(i=1,2,\cdots,n)$, 则D_i 类在集合 U 中出现的概率为$P_i=n_i/n$, 则计算集合 U 划分 d 个类的信息熵为$C_i(i=1,2,\cdots,c)$.

$$H(U)=-\sum_{i=1}^{d}P_i\log_2 P_i \tag{1}$$

假设分类属性集合 C 的属性值集合为 Values(C), C_{C_i}是集合 U 中属性 C 的值为C_i 的样本子集, 该子集实例个数为 t, 属于D_i 类的个数有$t_i(i=1,2,\cdots,t)$, 子集C_i在属性集合 C 出现的概率为$P_{t_i}=t_i/t$, 则属性集合 C 划分为 c 个类的信息熵为:

$$H(C_{C_i})=-\sum_{i=1}^{t}P_{t_i}\log_2 P_{t_i} \tag{2}$$

定义 1. 选择分类属性 C 后的实例训练样本集合 U 的信息熵为:

$$H(U,C)=\sum_{i=1}^{t}\left|C_{C_i}\right|/|U|H\left(C_{C_i}\right) \tag{3}$$

即在选择分类属性集合 C 后的信息熵为其每一个子集C_i的信息熵的加权和, 权值为子集C_{C_i}中值的个数占集合 U 的个数的比例.

定义 2. 属性 C 相对实例训练样本集合 U 的信息增益为:

$$Gain(U,C)=H(U)-H(U,C) \tag{4}$$

信息增益指的是因知道属性 C 后而导致集合 U 的信息熵下降, $Gain(U,C)$ 越小, 说明 $H(U)$ 下降得越快, $H(U,C)$ 所含的信息量就越大, 属性 C 就难以从众多的分类属性中分类出来. 所以, ID3 算法才选择信息增益最高的属性作为测试属性.

2 基于粗糙集属性约简的简化 ID3 算法

2.1 属性约简

定义 3. 设$D\subseteq C$, 如果 D 是独立的, 且 $IND(D)=IND(C)$, 那称 D 是等价关系族 C 的一个约简 (Reduct).

一个信息系统里面含有很多知识信息, 有些知识很可能是重复的或是没有必要, 也就是说这些知识对该信息系统是冗余的, 把多余的知识删掉后剩下的知识就是知识约简. 即知识约简还是与原来的信息系统保持有同样的分类能力的, 只不过是把一些没必要的知识删去而已, 从而使得分类时更加简便、精确.

定理 1[14]. 设$X\subseteq C$,

1) 如果$\forall x\in RED(X)$, 那:

$$SIG_{(RED(X)-\{x\})}(x)>0 \tag{5}$$

2) 如果$\forall x\in X-RED(X)$, 则:

$$SIG_{RED(X)}(x)=0 \tag{6}$$

从上面的命题可以看出, 知识约简中每一个元对约简中的其余元都是必不可少, 而约简外的其他元对于约简中的任何一元都是不必要, 即是可以删去.

定义 4. 设 C 和 D 为论域 U 上的等价关系, D 的 C 正域记作$POS_C(D)$, 定义为:

$$POS_C D=\cup\underline{C}X \tag{7}$$

其中, $X\in U/D$.

显然, 相对约简是两个属性之间的关系. $POS_C(D)$ 解释了知识U/C 的信息正确归类到属性 D 的等价类中的准确性.

定义 5. 设 C 和 D 为论域 U 上的等价关系族, $R\in P$, 如果:

$$POS_{IND(C)}(IND(D))=POS_{IND(C-\{R\})}(IND(D)) \tag{8}$$

就称 R 为 C 中 D 可省的, 反之就称 R 为 C 中 D 不可省[3]的. 如果 C 中任意一个关系 R 都是不可省的, 就称是 D 独立的, 也就是说 C 是相对于 D 独立的.

定义 6. 设$S\subseteq C$, 称 S 为 C 的约简, 当且仅当 S 是 C 的 D 独立子族, 且:

$$POS_S(D)=POS_C(D) \tag{9}$$

即称之为相对约简.

定理 2. C 的 D 核等于 C 的所有 D 约简的交:

$$CORE_D(C)=\cap RED_D(C) \tag{10}$$

定义 7. 令 $T=(U,A,V,f)$ 为一个信息系统, U 中元素的个数记为$|U|=n,|A|=m$, T 的分辨矩阵 M 定义为一个 n 阶对称矩阵[3], 其 i 行 j 列处的元素可定义为:

$$m_{ij}=\left\{a\in A|f(x_i,a)\neq f\left(x_j,a\right)\right\},\text{其中}i,j=1,\cdots,n. \tag{11}$$

其中, f 称为分辨函数, 使得m_{ij} 可以区分属性.

对于每一个$a\in A$, 指定布尔变量$\overline{a}$[2], 将 T 的 f 函数定义为一个 m 元布尔函数:

$$\rho(\overline{a_1},\overline{a_2},\cdots,\overline{a_m})=\wedge\left(\vee m_{ij}|1\leq j<i\leq n,m_{ij}\neq\emptyset\right) \tag{12}$$

其中, $\wedge$ 为合取, $\vee$ 为析取. 可以看出, 分辨函数 f 的析取范式中的每一个合取式对应一个约简, 约简的交集即为核.

2.2 化简算法

下面简化的公式是基于本章中的式 (1), 假设论域 U 中的训练样本集只有两类, 即决策属性集合 D 中只有两种类别, 称为正例与反例, 设其占的个数分别为 P 与 N, 分类属性集合 C 中对应的正例与反例的个数分别为p_i,n_i, 则公式可以写成:

$$\begin{aligned}H(U,C)&=\sum_{i=1}^{t}(n_i+p_i)/(N+P)\cdot[-p_i/(n_i+p_i)\log_2 p_i/\\&\quad(n_i+p_i)-n_i/(n_i+p_i)\log_2 n_i/(n_i+p_i)]\\&=\sum_{i=1}^{t}1/(N+P)[-p_i\log_2 p_i/(n_i+p_i)-n_i\log_2 n_i/(n_i+p_i)]\\&=-1/(N+P)\sum_{i=1}^{t}[p_i\log_2 p_i/(n_i+p_i)+n_i\log_2 n_i/(n_i+p_i)]\end{aligned}$$

由对数换底公式: $\log_a b=\log_c b/\log_c a$, 其中 a,b,c 为大于 0 且不等于 1 的常数.

故:

$$\begin{aligned}H(U,C)&=-1/(N+P)\cdot\sum_{i=1}^{t}\\&[p_i(\ln(p_i/n_i+p_i))/\ln 2+n_i(\ln(n_i/n_i+p_i))/\ln 2]\\&=-1/(N+P)\ln 2\sum_{i=1}^{t}[p_i\ln(p_i/n_i+p_i)+n_i\ln(n_i/n_i+p_i)]\end{aligned}$$

又由数学分析中的泰勒公式[15], 有:

$$\ln(1+x)=x-x^2/2+x^3/3+\cdots+(-1)^{n-1}x^n/n+o(x^n)$$

得$\ln(1-x)=-x-x^2/2-x^3/3+\cdots+(-1)x^n/n$.

当 x 趋于 0 时, $\ln(1-x)\approx -x-x^2/2$.

所以:

$$H(U,C)=-1/(N+P)\ln 2\cdot\sum_{i=1}^{t}[p_i\ln(1-n_i/n_i+p_i)+n_i\ln(1-p_i/n_i+p_i)]$$

$$\approx -1/(N+P)\ln 2\cdot\sum_{i=1}^{t}\left[\begin{array}{l}p_i\left(-n_i/n_i+p_i-{n_i}^2/2(n_i+p_i)^2\right)+\\ n_i\left(-p_i/n_i+p_i-{p_i}^2/2(n_i+p_i)^2\right)\end{array}\right]$$

$$\approx -1/(N+P)\ln 2\cdot\sum_{i=1}^{t}\left[-2n_ip_i/n_i+p_i-n_ip_i\left(n_i/(n_i+p_i)^2+p_i/(n_i+p_i)^2\right)/2\right]$$

$$\approx -1/(N+P)\ln 2\sum_{i=1}^{t}[-4n_ip_i/2(n_i+p_i)-n_ip_i/2(n_i+p_i)]$$

$$\approx -1/(N+P)\ln 2\sum_{i=1}^{t}-5n_ip_i/2(n_i+p_i)$$

$$\approx 5/(N+P)\ln 2\sum_{i=1}^{t}n_ip_i/n_i+p_i$$

因为$5/(N+P)\ln 2$为常数, 故信息熵简化公式可写为: $g=\sum_{i=1}^{t}n_ip_i/n_i+p_i$.

3 实例分析

本文中的例子摘自李雄飞等著者编写的第二版《数据挖掘与知识发现》, 详情见参考文献[3]. 下面以表 1 为例, 运用 ID3 算法构造决策树. 该表中一共有 14 个样本, 样本构成的集合为 U, 分类属性一共有 4 种, 决策属性只有 1 种, 且 PlayTennis 有两个不同的值{yes, no}, 也就是有两个不同类D_1, D_2.

表 1 PlayTennis 的训练样本集[3]

Day	Outlook	Temper -ature	Humi- dity	Wind	Play Tennis
D_1	Sunny	Hot	High	Weak	No
D_2	Sunny	Hot	High	Strong	No
D_3	Overcast	Hot	High	Weak	Yes
D_4	Rain	Mild	High	Weak	Yes
D_5	Rain	Cool	Normal	Weak	Yes
D_6	Rain	Cool	Normal	Strong	No
D_7	Overcast	Cool	Normal	Strong	Yes
D_8	Sunny	Mild	High	Weak	No
D_9	Sunny	Cool	Normal	Weak	Yes
D_{10}	Rain	Mild	Normal	Weak	Yes
D_{11}	Sunny	Mild	Normal	Strong	Yes
D_{12}	Overcast	Mild	High	Strong	Yes
D_{13}	Overcast	Hot	Normal	Weak	Yes
D_{14}	Rain	Mild	High	Strong	No

为了便于计算, 把表 1 转化为 PlayTennis 训练集数值化数据, 如表 2.

表 2 PlayTennis 训练集数值化数据

U	Outlook	Temperature	Humidity	Wind	PlayTennis
x_1	0	0	0	0	0
x_2	0	0	0	1	0
x_3	1	0	0	0	1
x_4	2	1	0	0	1
x_5	2	2	1	0	1
x_6	2	2	1	1	0
x_7	1	2	1	1	1
x_8	0	1	0	0	0
x_9	0	2	1	0	1
x_{10}	2	1	1	0	1
x_{11}	0	1	1	1	1
x_{12}	1	1	0	1	1
x_{13}	1	0	1	0	1
x_{14}	2	1	0	1	0

第 1 步. 由决策表 2 PlayTennis 训练集数值化数据, 可得:

$$U=\{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{12},x_{13},x_{14}\}$$

$$U/C=\left\{\begin{array}{l}\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\},\{x_6\},\{x_7\},\\ \{x_8\},\{x_9\},\{x_{10}\},\{x_{11}\},\{x_{12}\},\{x_{13}\},\{x_{14}\}\end{array}\right\}U/D=$$
$$\{\{x_1,x_2,x_6,x_8,x_{14}\},\{x_3,x_4,x_5,x_7,x_9,x_{10},x_{11},x_{12},x_{13}\}\}$$

$$k=|POS_CD|/|U|=1$$

故该决策表是一致决策表, 即集合 D 依赖于集合 C.

第 2 步. 画出决策表的分辨矩, 见表 3

第 3 步. 通过分辨函数求出属性的约简.

由画出决策表的分辨矩, 见表 3, 得分辨函数为:

$$f=O(O\vee W)(O\vee T)(O\vee T\vee W)(O\vee T\vee H)$$
$$(O\vee T\vee H\vee W)(T\vee H\vee W)W(T\vee H)\wedge(T\vee W)$$
$$(O\vee H)(H\vee W)(O\vee H\vee W)=O(T\vee H\vee W)W(T\vee H)$$
$$=OW\wedge(T\vee H)=OWT\vee OWH$$

因此, 决策表的核为: $OWT\cap OWH=OW$

即说明 Outlook 与 Wind 这个属性在做决策分类时是必不可少, 该系统的分类属性只需要 Outlook、Wind、Temperature 或者 Outlook、Wind、Humidity 这 3 个就可以进行归类.

第 4 步. 选择 Outlook、Wind、Temperature 这 3 个属性进行决策, 用信息熵简化公式 g 计算.

Outlook: $g_o=2*3/(2+3)+4*0/(4+0)+3*2/(3+2)=6/5$同理, 可求得: $g_w=27/8, g_T=37/12$.

显然, Outlook 的信息熵最小, 故取其作为根节点, 画出部分树如图 1.

第 5 步. 在属性 Outlook 的条件下继续计算分类属性的信息熵, 选择最小者作为节点.

同理, 当 Outlook 取值为 Sunny 时, 有:

Wind: $g_{\text{Sunny}_W} = 7/6$.

Temperature: $g_{\text{Sunny}_T} = 1/2$.

故在树的左边第二节点选择属性 Temperature.

表 3 决策表 2 的分辨矩阵

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1														
2	\													
3	*O*	*OW*												
4	*OT*	*OTW*	\											
5	*OTH*	*OTHW*	\	\										
6	\	\	*OTHW*	*THW*	*W*									
7	*OTHW*	*OTH*	\	\	\	*O*								
8	\	\	*OT*	*O*	*OTH*	\	\							
9	*TH*	*THW*	\	\	\	*OW*	\	*TH*						
10	*OTH*	*OTHW*	\	\	\	*TW*	\	*OH*	\					
11	*THW*	*TH*	\	\	\	*OT*	\	*HW*	\	\				
12	*OTW*	*OT*	\	\	\	*OTH*	\	*OW*	\	\	\			
13	*OH*	*OHW*	\	\	\	*OTW*	\	*OTH*	\	\	\	\		
14	\	\	*OTW*	*W*	*THW*	\	*OTH*	*OW*	*OTHW*	*HW*	*OH*	*O*	*OTHW*	

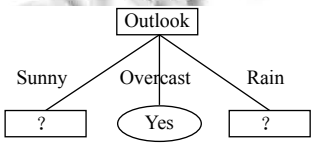

图 1 部分决策树

以此类推, 直到当前节点的训练样本实例是属于同一类时就可以结束计算了, 得出一棵决策树如图 2.

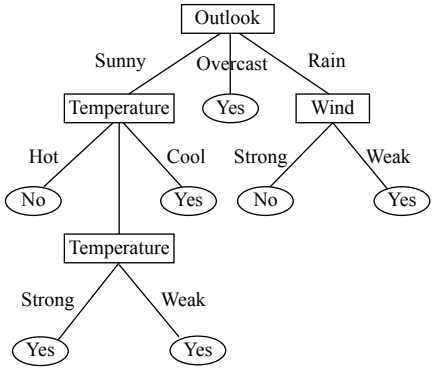

图 2 决策树 1

如果在第三步中选择 Outlook、Wind、Humidity 这 3 个属性的话, 得到决策树结果跟用传统 ID3 算法构造的一样, 如图 3.

4 仿真实验与结果分析

4.1 仿真实验环境与实验内容

本实验涉及的设备、语言等具体环境如表 4.

为了验证改进算法的运行时间比 ID3 的少, 本文用 C++语言编写程序进行实验, 实验数据主要来源于 UCI Machine Learning Repository, 选取了 Balance Scale Data Set 库, 网址是: http://archive.ics.uci.edu/ml/datasets.html.

决策属性为 Class Name: Yes, No, 分类属性有 4 种, 如表 5.

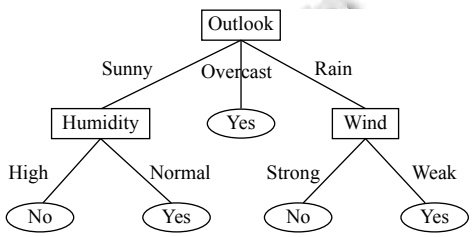

图 3 决策树 2

表 4 实验环境

运行设备	ASUS Windows 8.1 64bits
开发语言	C++
运行软件	Microsoft Visual C++ 6.0

表 5 实验内容的分类属性与种类

分类属性	种类
Left-Weight	1, 2, 3, 4, 5
Left-Distance	1, 2, 3, 4, 5
Right-Weight	1, 2, 3, 4, 5
Right-Distance	1, 2, 3, 4, 5

这个数据集是为了模拟心理学实验结果而产生. 本文中选取的每个例子被分类为平衡秤尖端在右侧或

是在向左端, 实验中分别标记为 Yes, No. 属性是左重量, 左距离, 右重量和右距离.

该实验用 C++编程的流程如图 4.

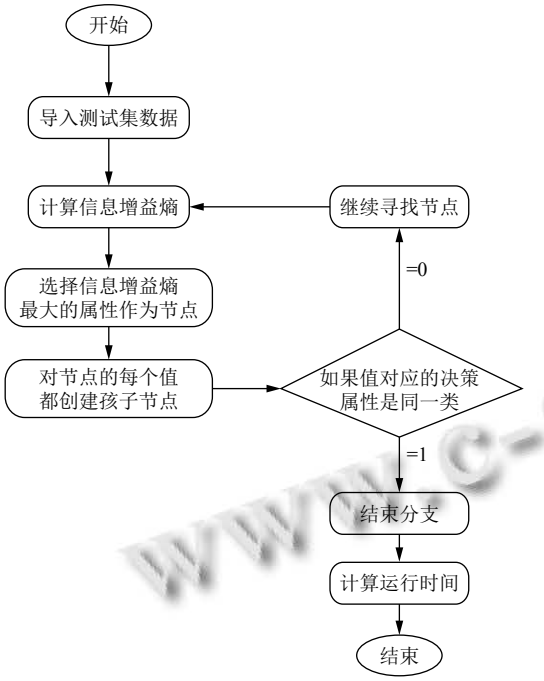

图 4 编程流程图

4.2 实验结果分析

4.2.1 算法的正确性与可行性

ID3 算法用的信息熵公式:

$$H(U) = -p_1\log_2 p_1 - p_2\log_2 p_2 \tag{13}$$

其中, p_1, p_2分别指的是实例训练集合中正例与反例所占的比例.

基于粗糙集方法的属性约简化简算法用的信息熵公式为:

$$H(U') = p_i * n_i/(p_i + n_i) \tag{14}$$

其中, p_i, n_i 分别指的是实例训练集合中正例与反例的个数.

由此可画出式 (14) 与式 (15) 的函数图像如图 5.

由图 5 可知, ID3 算法与基于粗糙集属性约简的简化 ID3 算法的熵的范围都是 0 到 1, 但是改进的算法的熵的变化幅度比 ID3 的大. 基于粗糙集属性约简的简化 ID3 算法运算复杂度要比 ID3 算法的要低得多, 省去了 log 对数的计算, 并且保持了跟传统 ID3 算法一样的准确率. 因此, 改进的算法并没有改变熵函数的性质, 可见其的正确性与可行性, 并且扩大了熵的变化幅度.

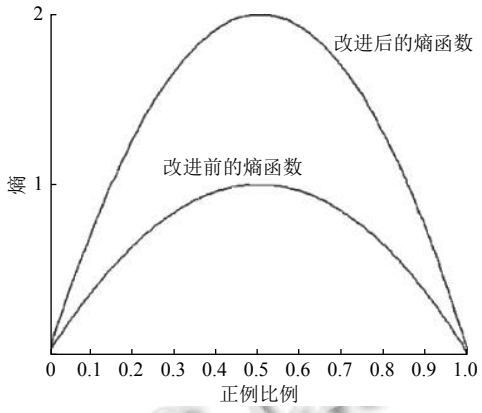

图 5 熵的函数图像

4.2.2 算法的优势

通过实验可知, 基于粗糙集方法的属性约简化简算法的计算时间比 ID3 少, 具体如表 6.

从上述仿真实验以及分析可以得出以下结论:

1) 基于粗糙集方法的属性约简化简算法的熵函数性质并没有改变, 继承了 ID3 的所有优点;

2) 基于粗糙集方法的属性约简化简算法的计算时间比 ID3 的少, 这无疑是一个优势.

表 6 实验结果

训练集合样本个数	测试集合样本个数	ID3 计算时间 (ms)	改进后的算法计算时间 (ms)	两者的计算时间之比
576	10	88	71	1.2394
576	30	156	125	1.2480
576	50	310	260	1.1923
576	100	404	352	1.1477
576	300	496	423	1.1726
576	500	528	453	1.1656
576	576	696	562	1.2384

5 结论

本文先通过文献综述总结了决策树 ID3 的历史发展过程, 概述了当前学者研究改进及优化决策树 ID3 算法的现状和结论. 本文通过实例分析与实验验证, 基于粗糙集属性约简的简化 ID3 算法的优势之一是删掉了冗余的知识, 是决策系统更加精炼简洁, 能快速找出关键的分类属性; 优势之二是保持了 ID3 的一切优点, 与 ID3 有同样的准确率与精度; 优势之三是选取测试属性的计算公式中没有复杂的对数函数, 只有简单的加法、乘法和除法, 经实验证明, 运算时间比 ID3 的要少. 缺点是在求属性约简时, 规模过大的话, 用分辨函数求解的难度就会变大.

总的来看, 本文处于研究阶段并没有把结论应用到社会生活实例中去, 所以本文还是存在不足之处, 还是有很多值得再去深入研究、探索的地方, 未来的研究方向以下这两点: 1) 本文提出的基于粗糙集的决策树 ID3 算法相对 ID3 来说确实是有优势, 但在求属性的约简过于繁琐, 所以下一步的研究是寻求更加简单的方式去求属性的约简, 完善基于粗糙集方法的属性约简化简决策树算法. 2) 由于时间及其能力不足等等客观条件的限制, 本文没有将改进的算法运用到更多的实际生活例子中去并用于预测, 因此该论文得到发展的话, 应该更重视研究把此算法改得更加合理性、科学性、可行性.

参考文献

1 Agrawal R, Imielinski T, Swami A. Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering, 1993, 5(6): 914–925. [doi: 10.1109/69.250074]

2 孙道远. 决策树 ID3 算法中引入简单工厂模式的设计研究. 德州学院学报, 2018, 34(2): 61–64. [doi: 10.3969/j.issn.1004-9444.2018.02.016]

3 李雄飞, 董元方, 李军. 数据挖掘与知识发现. 2 版. 北京: 高等教育出版社, 2010. 67–175.

4 黄爱辉, 陈湘涛. 决策树 ID3 算法的改进. 计算机工程与科学,2009,31(6):109–111.[doi:10.3969/j.issn.1007-130X.2009.06.033]

5 杨霖, 周军, 梅红岩, 等. ID3 改进算法研究. 软件导刊, 2017, 16(8): 21–24.

6 王小巍, 蒋玉明. 决策树 ID3 算法的分析与改进. 计算机工程与设计, 2011, 32(9): 3069–3072, 3076.

7 Li JF, Lei JH, Zhao XX, *et al*. An improved ID3 algorithm. Applied Mechanics and Materials, 2013, 444–445: 723–727. [doi: 10.4028/www.scientific.net/AMM.444-445.723]

8 Jiang MH, Luo XS. Classification of student achievement using ID3 algorithm. Applied Mechanics and Materials, 2012, 220–223: 2540–2545. [doi: 10.4028/www.scientific.net/AMM.220-223.2540]

9 Shao XY, Zhang GJ, Li PG, *et al*. Application of ID3 algorithm in knowledge acquisition for tolerance design. Journal of Materials Processing Technology, 2001, 117(1–2): 66–74. [doi: 10.1016/S0924-0136(01)01016-0]

10 李建, 付小斌, 吴媛媛. 基于优化 ID3 的井漏类型分类算法. 计算机工程, 2019, 45(2): 290–295.

11 于天佑, 张楠, 岳晓冬, 等. 基于多特定类的序决策表下近似约简. 计算机科学, 2019, 46(10): 242–251. [doi: 10.11896/jsjkx.180901781]

12 邬阳阳, 郭文强, 汤建国, 等. 几类拓展粗糙集模型属性约简研究综述. 宜宾学院学报. https://doi.org/10.19504/j.cnki.issn1671-5365.20190531.001. [2019-09-07].

13 尹继亮. 基于对象相似度的属性约简研究[硕士学位论文]. 烟台: 烟台大学, 2019.

14 史开泉, 崔玉泉. S-粗集与粗决策. 北京: 科学出版社, 2006. 8–15.

15 华东师范大学数学系. 数学分析. 4 版. 北京: 高等教育出版社, 2010. 137–145.