

DOI:10.3876/j.issn.1000-1980.2013.02.016

非相容表决策树构建算法

刘 军

(南京工业大学电子与信息工程学院,江苏 南京 210009)

摘要: 针对非相容表构建决策树中属性约简计算复杂度较高的问题,基于粗糙集分辨关系和粒计算理论,建立一种对相容表和非相容表通用的构建决策树算法。该算法首先根据非相容表的逻辑关系式证明其可经过变量替代等价于相容表;然后根据等价表的特点,以属性粒为基本单位,用属性粒的可分辨量、属性粒相对于决策属性的类别数量、决策表中相同行的数量这 3 个基本参数确定属性的优劣,并以最优属性为划分属性自上向下直接构建决策树。实例分析结果表明,应用该算法构建决策树具有简洁、有效性。

关键词: 粗糙集;决策树构建;非相容表;属性粒

中图分类号: TP18 **文献标志码:** A **文章编号:** 1000-1980(2013)02-0177-07

Algorithm for decision tree construction for incompatible table

LIU Jun

(College of Electronics and Information Engineering, Nanjing University of Technology,
Nanjing 210009, China)

Abstract: To overcome the difficulties in complex attribute reduction calculation during decision tree construction for incompatible tables, a general algorithm for constructing a decision tree suitable for compatible and incompatible tables is put forward based on the rough set of resolution relations and the granulation computing theory. According to the logical relationship of the incompatible table, this algorithm proves that the table created by the substitution of variables can be equivalent to a compatible table. Then, according to the features of the equivalent table, with the attribute granulation being regarded as the basic unit, three basic parameters, the resolution volume of attribute granulation, the class of attribute granulation relative to the decision attribute, and the number of same lines in the decision table, are used to determine the importance of the attribute and construct the decision tree from top to bottom with the best attribute. Experiments and comparison show that this algorithm for constructing the decision tree is simple and efficient.

Key words: rough set; decision tree construction; incompatible table; attribute granulation

当前,决策表建树算法大多是针对相容表的,为解决决策表的属性数量有限等不确定性问题,学者引入了非相容表。非相容表决策树构建算法需要解决 2 个主要问题,即如何将非相容表等价为相容表 and 如何对这类等价表构建决策树。在非相容表等价为相容表方面,秦川等^[1]对相容表的 Skowron 可辨识矩阵的定义进行改进,使其对非相容表也能进行属性约简;叶东毅^[2]用粗糙集的 P 近似精度 $\gamma_p(U, D)$ 进行判别。上述学者的思路皆以属性性质判别相容关系为前提,这种全表的列向(属性)约简算法不仅要判定相容性,而且需要进行属性约简,因此运算量比较大。为此,笔者采用行向(记录)的逻辑关系判定相容关系,并将非相容行经过变量替代后等价于相容行,最终得到等价相容表。在如何构建决策树方面,基于信息熵的决策树算法往往存在子树重复的情况,降低了判定树的使用效果^[3]。近年来基于粗糙集的决策树构建是研究的热点^[3-4],属性约简是粗糙集理论的核心^[4]。由于属性约简的结果直接影响决策树构建的质量,所以往往希望

收稿日期:2012-06-06

基金项目:国家自然科学基金(60673185);教育部留学回国人员科研启动基金(200711108)

作者简介:刘军(1977—),女,江苏南京人,讲师,硕士,主要从事数据挖掘研究。E-mail:1055030391@qq.com

求得最小约简集,然而要计算出最小约简集是一个 NP 难题^[5]。目前常用的属性约简算法是基于分辨矩阵的方法^[6]和基于正区域的方法^[7],其中心思想都是利用粗糙集的属性约简技术来预先去除与构建决策树无关的属性(属性约简)^[8],但这些方法的属性约简计算复杂度高,为解决此问题,笔者拟采用不进行属性约简而直接构建决策树的方式,即以粗糙集分辨关系^[9]和粒计算^[10]为理论基础,给出一种针对非相容或相容表构建决策树的通用算法。该算法仅以条件属性集 A 中单一属性 A_i 对决策属性集 D 的分辨关系为前提,将分辨能力较强、击中数量最大且可尽快分出叶结点的属性作为划分属性,以此递进划分逐级完成决策树的构建,将相同行的数量(击中数量)作为属性优选的一个重要参数,这样能保证生成的决策树上层多为击中量大的叶结点且可尽量减少树的结点深度。本算法具有以下特点:

- a. 运用多值逻辑关系式证明单一条件下得出多种不同结论的不相容问题,可以将多结论以一个逻辑变量替代,将其等价于单一条件单一结论的相容问题进行处理。
- b. 针对替代性相容表的类型多变问题,提出一种基于粗糙集分辨关系的决策树生成算法。该算法不以常规的属性约简求核属性方法为出发点,而是以单一属性 A_i 对 D 的分辨关系为前提确定优选属性; A_i 按其取值划分为若干部分(属性粒),并将属性粒作为一个局部参数;该属性粒对应于 D 的决策属性粒数量(类别数量)作为一个全局参数;将表中相同行的数量作为属性粒的辅助参数,由这 3 个参数确定该属性粒的优劣。 A_i 中各属性粒的累加和即为 A_i 的优劣度,最优者即为划分属性。
- c. 从实际出发,将决策表中相同行(包括逻辑变量替代后的相同行)作为一类辅助参数参与属性粒的择优。相同行越多表示这类情况出现的概率越大,则应该增加该行的权重。

1 决策树构建理论和算法

1.1 理论

设 $S=(U, A, D, V, F)$ ^[9] 是一决策表,其中, $U=\{x_1, x_2, \cdots, x_n\}$ 是论域, A 为条件属性集, D 为决策属性集, V 为属性值集合, F 是 $U \times (A \cup D) \rightarrow V$ 的映射。根据 Lukasiewicz 的多值逻辑定义^[11], S 是多值逻辑的真值表, S 中的每一行都是多值逻辑的一个最小项,由最小项的相容与不相容关系可以确定如何将不相容关系替代为相容关系。

定理 1 设多值逻辑不相容最小项 $m_i=p, m_j=q$ 和 $m_k=r$ 。最小项 m_i, m_j 可相容的充要条件是 $m_k=r$ (其中 i, j 和 k 是最小项下标, p, q 和 r 为逻辑函数值, $m_i=m_j=m_k, p \neq q, r=p+q$)。

证明 由 $m_i \vee m_j = p \vee q$ (因 $m_i=m_j=m_k, p \neq q \rightarrow m_k=p \vee q$)。

由独立命题逻辑运算^[12]得概率公式:

$$P(p \vee q) = P(p) + P(q) - P(p) \times P(q) \tag{1}$$

因 p 和 q 是独立事件,且 $r=p+q$,因此 r 也是一个独立事件,则

$$P(p \vee q) = P(p) + P(q) = P(r) \tag{2}$$

式 $m_k=r$ 成立,得证。

定理 1 说明任何不相容的决策表,将其不相容的多个最小项经独立替代后可以转换为一个独立最小项,使原决策表相容,从而使不相容表构建决策树转换为相容表构建决策树。构建决策树完成后将先前替代的独立事件 r (叶结点)逆替代为 p 和 q (叶结点)即可。

当前,决策树构建方法首先需要对 S 进行属性约简^[13]。由于 S 的情况各异(尤其是经替代后的 S),往往会出现从极端紧密(每个属性都是核属性)至极端松散(有多个核集)的各种情况。极端紧密不需要属性约简,而极端松散则会出现多个可选择的属性约简集,其他情况介于二者之间,所以需要一种可处理各类 S 的统一算法。

定义 1^[10] 在 S 中,令 $A_i \in A, v \in V, (A_i, v)$ 或 A_{iv} 为 S 上的原子公式, $m(A_{iv})$ 表示 U 上所有满足 $f(A_i)=v$ 的个体集合,其中 $m(*)$ 为意义函数符,则称二元对 $(A_i, m(A_{iv}))$ 为 S 上的基本粒。

根据定义 1,基本粒 $(A_i, m(A_{iv}))$ 主体是原子公式 (A_i, v) ,即 (A_i, v) 表示属性 A_i 取值为 v 时 $u \in U$ 的行集,所以在本文中 (A_i, v) 称为 S 的属性粒,而 (D, d_i) 称为 S 的决策属性粒,其中 $d_i \in D$ 。属性粒是属性的最小可划分单位,以其作为分辨关系的基本单位可以准确地确定属性的分辨强度。

定义 2 设属性 $A_i \in A, v \in A_i$ 。 A_i 的属性同值量定义为 $Y_{A_{iv}} = \text{count}(x_i \in U \mid (A_i, v))$ 。 $Y_{A_{iv}}$ 是属性 A_i 取值为

v 时所对应 S 的行数, $Y_{A_{iv}}$ 是独立参数。

定义 3 将 S 的完全相同行只用一行表示, 同时新增一列属性 n 记录该相同行的个数, 称 n 为属性同值量系数。

在设有 n 的 S 中, $Y_{A_{iv}}$ 和 n 配合使用, 即 (A_i, v) 所涉及 n 之累加和记为 $N_{A_{iv}}$ 。对 S 中未设 n 列的情况, 则默认 $n=1$ 。

例如: 实例分析中表 3 中未设 n 列, 属性 a 中有 2 种取值 1 和 2, 其对应 Y_{av} 分别是: $Y_{a1}=6; Y_{a2}=15$; 表 1 中设有 n 列, 属性 e 中有 2 种取值 0 和 1, 其对应 N_{ev} 分别是: $N_{e0}=10+30=40; N_{e1}=10+40+90+60=200$ 。

定义 4 设属性子集 $(\{A_i \cup D\} \mid (A_i, v), d_i \in D)$ 。 $X_{A_{iv}} = \text{count}(x_i \in U \mid (A_i, v), d_i \in D)$ 是当 $A_i = v$ 时相应行集中 D 的种类个数, 称 $X_{A_{iv}}$ 为 (A_i, v) 相对 D 的类别数。 $X_{A_{iv}}$ 是 (A_i, v) 对应 D 之间的分辨关系参数, 表示 (A_i, v) 能分辨 D 的程度, 是 (A_i, v) 相对于 (D, d_i) 的关联参数。

例如: 表 1 中, 属性 e 的 2 种取值 0 和 1, 其对应 X_{ev} 分别是: $X_{e0}=1 \mid (D=1); X_{e1}=3 \mid (D=0, 1, 2)$ 。

1.2 算法

每个属性 A_i 由若干 (A_i, v) 组成, 所以可以先确定 A_i 中每个 (A_i, v) 的分辨能力, A_i 中各 (A_i, v) 分辨能力的累加和即为 A_i 的分辨能力。 求出 A 中每个 A_i 的分辨能力后, 再从中选出分辨能力最强者作为划分属性, 由此由上而下构建决策树。

决定 (A_i, v) 分辨能力的关键参数是 $N_{A_{iv}}$ 和 $X_{A_{iv}}$ 。 从 1.1 节分析可知, (A_i, v) 的 $N_{A_{iv}}$ 值越大且 $X_{A_{iv}}$ 值越小, 则其分辨能力越强。 所以依据 $N_{A_{iv}}$ 和 $X_{A_{iv}}$ 可以推导出判别 (A_i, v) 分辨能力强弱的一般式: 若 $X_{A_{iv}}=1$, 则经 1 次结点划分, $N_{A_{iv}}$ 即可分辨出 $d_i \in D$ 中的 $N_{A_{iv}}$ 行, 即 $N_{A_{iv}}/X_{A_{iv}}=N_{A_{iv}}/1$, 得 1 个叶结点; 若 $X_{A_{iv}}=2$, 1 次结点划分后, 要在之后的 2 个分区内完成对 $d_i \in D$ 的分辨, 至少要经 2 次结点划分, $N_{A_{iv}}$ 才可 2 次分辨完 $d_i \in D$ 的 $N_{A_{iv}}$ 行, 即 $N_{A_{iv}}/X_{A_{iv}}=N_{A_{iv}}/2$, 得 2 个叶结点; 若 $X_{A_{iv}}=3$, 1 次结点划分后, 要在之后的 3 个分区内对 $d_i \in D$ 进行分辨, 至少需要经过 3 次结点划分, $N_{A_{iv}}$ 才可 3 次分辨出 $d_i \in D$ 的 $N_{A_{iv}}$ 行, 即 $N_{A_{iv}}/X_{A_{iv}}=N_{A_{iv}}/3$, 得 3 个叶结点……依此类推, 可以得到 (A_i, v) 对 $d_i \in D$ 的分辨度的一般式: $U_{A_{iv}} = N_{A_{iv}}/X_{A_{iv}}$, 属性 A_i 对 D 的分辨度算式:

$$U_{A_i} = \sum U_{A_{iv}} \mid (\forall v \in A_i) = \sum \frac{N_{A_{iv}}}{X_{A_{iv}}} \mid (\forall v \in A_i) \tag{3}$$

而 $U_{A_i} (i=1, 2, \dots, n)$ 中最大者为划分属性: $M = \max(U_{A_i}) (i=1, 2, \dots, n)$ 。

在当前表 S 中, 用 M 进行划分生成 1 个枝结点且得若干子表, 再对各子表求其各自的 M , 直到当前子表中 M 中只有 1 种 $d_i \in D$ 值, 则生成 1 个叶结点, 这是一个逐级择优划分的递归算法:

```
CreateDt(A, D, A_i) { //程序入口
if( count(D) = count(A_i) ) { //判断 S 中 D 是否只有一类值
CreateNode(A_i, D); return } //建立 1 个叶结点且回退
for( ; ∀ A_i ∈ A; ) { //对 A 中的每个属性循环
for( ; ∀ v ∈ A_i; ) { //对 A_i 中的每种值循环
U_{A_i} += ( N_{A_{iv}}/X_{A_{iv}} ) //求 A_i 的分辨度 U_{A_i}
if( U_{A_i} > M ) M = U_{A_i}; } //求划分属性 M
CreateNode(M, D) //用 M 构建 1 个枝结点
for( ; ∀ v ∈ M; ) { //对 M 中的每种值循环
U = { x_i } | (v ∈ M) //求 M 中等于 v 值的行
{ A, D } | U = { x_i } //按 U 对 { A, D } 划分
A -= M //在 { A } 中去除属性 M
CreateDt(A, D, M) } } //递归调用
其中 CreateNode(A, D) { //相容-非相容叶结转换子程序
Select D_x from (A, D) //从原始表(A, D)中获得非相容决策表每个决策属性值 D_x
for( ; ∀ d_i ∈ D_x; ) CreateLeafNode(d_i) //为每个决策属性值建立 1 个叶结点 d_i
```

2 实例分析与比较

2.1 实例分析

文献[14]中表1是一个非相容决策表,将非相容行中的 $D=1 \vee 0$ 用2替代,同时增加表示表中相同行数量的列 n ,见表1。

首层划分属性 M 的选择:

$$U_a = \sum U_{av} \mid (\forall v \in a) =$$
$$\sum \frac{Na_v}{Xa_v} \mid (\forall v \in a) =$$
$$\sum U_{av}(v=0,1) =$$
$$\frac{Na_0}{Xa_0} + \frac{Na_1}{Xa_1} =$$
$$\frac{130}{3} + \frac{110}{2} \approx 98 \tag{4}$$

同理, $U_b \approx 107, U_c = 80, U_d \approx 113, U_e \approx 107, U_f = 120$ 。

由 $M = \max(U_{A_i}) \rightarrow U_f = 120 \quad (A_i = a, b, c, d, e, f)$ 。

经 M 划分后,在 $f=0(U=Z_1, Z_2, Z_3)$ 的子表中: $U_a = 10/1+50/1=60, U_b = 50, U_c = 30, U_d = 60, U_e = 35, U_a$ 和 U_d 都是 M (等价)。可任选一个再次划分后得到叶结点。在 $f=1(U=Z_4, Z_{56}, Z_{78})$ 的子表中: $U_a = 120/2+60/1=120, U_b = 90, U_c = 90, U_d = 90, U_e = 180$ 。 U_e 是 M ,选其划分后得叶结点。决策树(以磁盘管理格式)表示如下:

$$[f]$$
$$f=0$$
$$[a]:a=0 \rightarrow D=0; a=1 \rightarrow D=1$$
$$f=1$$
$$[e]:e=0 \rightarrow D=1; e=1 \rightarrow D=2 \rightarrow \{D=0(47\%) \vee D=1(53\%)\}$$

当 $(f=1) \wedge (e=1)$ 时, $D=0$ 有47%可能,而 $D=1$ 有53%可能。这很好地解决了非相容表构建决策树的一个难题。从属性约简角度,属性重要程度的顺序为 f, e, a 或 d 。对照文献[14]基于正域约简集 $\{a, e, f\}, \{b, e, f\}$ 及 $\{d, e, f\}$ 和基于分布约简集 $\{a, e, f\}$ 可以看出,算法的属性择优是准确的。

文献[15]表1中 $U=4$ 与 $U=8$ 等价, $U=1$ 与 $U=5, 11, 20$ 等价, $U=2$ 与 $U=15$ 等价,而 $U=12$ 与 $U=3, 6, 14, 19$ 非相容,将 $D=1 \vee 0$ 用2替代,同时增加表示表中相同行数量的列 n ,得表2。

首层划分属性 M 的选择:

$$U_{c1} = \sum U_{c1v} \mid (\forall v \in c_1) =$$
$$\sum \frac{N_{c1v}}{X_{c1v}} \mid (\forall v \in c_1) =$$
$$U_{c1v}(v=0,1) =$$
$$\frac{N_{c10}}{X_{c10}} + \frac{N_{c11}}{X_{c11}} =$$
$$\frac{10}{3} + \frac{10}{2} \approx 8.3 \tag{5}$$

同理, $U_{c2} = 12.0, U_{c3} = 10.0, U_{c4} = 8.0, U_{c5} \approx 7.7$ 。

由 $M = \max(U_{A_i}) \rightarrow U_{c2} = 12.0 \quad (A_i =$

$c_1, c_2, c_3, c_4, c_5)$ 。

经 M 划分,在 $c_2=0(U=u_1, u_4, u_{10}, u_{13})$ 子表中直接得 $c_2=0 \rightarrow D=0$;在 $c_2=1(U=u_2, u_7, u_9, u_{12}, u_{16}, u_{17},$

表1 非相容表替代后的决策数据

Table 1 Decision data after substitution in incompatible table

U	a	b	c	d	e	f	D	n
Z_1	0	0	1	0	1	0	0	10
Z_2	1	1	1	1	1	0	1	40
Z_3	1	0	1	1	0	0	1	10
Z_4	0	0	1	0	0	1	1	30
Z_{56}	0	0	1	0	1	1	2	90
Z_{78}	1	0	1	0	1	1	2	60

表2 替代后的决策数据

Table 2 Decision data after substitution

U	c_1	c_2	c_3	c_4	c_5	D	n
u_4	0	0	0	0	1	0	2
u_{17}	0	1	0	0	0	0	1
u_9	0	1	1	1	0	1	1
u_{12}	0	1	1	0	0	2	5
u_{16}	0	1	1	1	1	1	1
u_1	1	0	0	1	0	0	4
u_{10}	1	0	0	0	0	0	1
u_{13}	1	0	0	0	1	0	1
u_7	1	1	0	1	1	1	1
u_2	1	1	1	0	0	1	2
u_{18}	1	1	1	1	1	1	1

u_{18})子表中得 $c_4=1 \rightarrow D=1$,对 $c_4=0$ 继续划分后最终得决策树:

$[c_2]:c_2=0 \rightarrow D=0$
 $c_2=10$
 $[c_4]:c_4=1 \rightarrow D=1$
 $c_4=0$
 $[c_3]:c_3=0 \rightarrow D=1$
 $c_3=1$
 $[c_1]:c_1=1 \rightarrow D=1;$
 $c_1=0 \rightarrow D=2 \rightarrow \{D=0(20\%) \vee D=1(80\%)\}$

属性的重要程度顺序是 $c_2(U_{c_2}=10.0), c_3(U_{c_3} \approx 8.3), c_1(U_{c_1}=12.0), c_4(U_{c_4}=8.0)$ 和 $c_5(U_{c_5} \approx 7.7)$ 。而文献[15]所述的顺序为 $c_1(U_{c_1}=0.1), c_3(U_{c_3}=0.05), c_4(U_{c_4}=0.05), c_2(U_{c_2}=0)$ 和 $c_5(U_{c_5}=0)$,若按文献[15]的结果应首选 c_1 作为 M ,则得4层树(图略):第1层1结点0树叶、第2层2结点3树叶、第3层1结点1树叶、第4层1结点2树叶。文献[15]的结果与本例树相比较差(其树叶多在下层)。由此可知,属性约简是侧重哪个属性最不可少(判定核属性),而本文所构建决策树算法则侧重哪个属性最能尽快生成树叶且逐渐包括各核属性。

2.2 结果比较

在解决了决策表非相容问题的前提下,可以直接对各类相容表构建决策树,因算法采用单属性与决策属性的逻辑关系进行决策树构建,所以算法简洁、有效。文献[16]中表1经属性(值)替代后如表3所示。

表3 属性(值)替代后的汽车数据
Table 3 Car data after substitution

U	a	b	c	d	e	f	g	h	i	s	U	a	b	c	d	e	f	g	h	i	s
1	2	4	2	1	2	2	3	2	2	2	12	1	4	1	1	1	1	2	1	2	3
2	2	4	1	1	2	1	3	1	2	2	13	2	4	1	2	2	1	2	1	2	2
3	2	4	1	1	2	2	3	1	2	2	14	1	4	2	1	1	2	3	1	2	3
4	2	4	2	1	2	2	3	1	1	3	15	1	4	2	2	1	1	1	1	2	3
5	2	4	1	1	2	1	2	1	2	2	16	2	4	2	1	2	1	3	1	2	2
6	2	4	1	2	2	1	2	2	3	1	17	2	4	1	1	2	1	3	2	2	2
7	2	4	1	1	2	1	3	1	3	1	18	2	4	1	1	2	1	3	2	2	2
8	1	4	1	2	1	2	1	1	1	3	19	1	4	1	1	1	2	2	1	2	3
9	2	4	1	2	1	2	1	1	2	2	20	2	4	1	1	1	2	2	1	2	3
10	2	4	1	2	1	2	2	2	2	2	21	2	4	1	2	1	2	2	1	2	2
11	1	4	1	1	1	2	1	1	1	3											

首层划分属性 M 的选择:

$$U_a = \sum U_{av} \mid (\forall v \in a) = \sum \frac{Y_{av}}{X_{av}} \mid (\forall v \in a) =$$
$$\sum U_{av}(v=1,2) = \frac{Y_{a1}}{X_{a1}} + \frac{Y_{a2}}{X_{a2}} = \frac{6}{1} + \frac{15}{3} = 11$$

(6)

同理, $U_b=7, U_c \approx 10.3, U_d=7, U_e \approx 8.7, U_f \approx 8.8, U_g \approx 7.7, U_h \approx 7.8, U_i=13$ 。

$M=\max(U_{A_i}) \rightarrow U_i=13(A_i=a,b,c,d,e,f,g,h,i)$ 。

经 M 划分后,本层得2个树叶,而在 $i=2$ 的子表中继续划分后最终得决策树:

$[i]:i=1 \rightarrow s=3(3); i=3 \rightarrow s=1(2)$
 $i=2$
 $[e]:e=2 \rightarrow s=2(8)$
 $e=1$
 $[a]:a=1 \rightarrow s=3(4)$
 $a=2$
 $[d]:d=2 \rightarrow s=2(3); d=1 \rightarrow s=3(1)$

与文献[16]中图1相比,同样是4层树,本算法至第2层已分辨出13行,而文献[16]中图1只分辨出9行。若按文献[16]改进算法将表1视为只有一个属性*i*(质量)和一个决策属性*s*(里程)的非相容表进行构建决策树,按1.2节算法可得决策树:

$[i]: i=1 \rightarrow s=3(100\%); i=3 \rightarrow s=1(100\%)$
 $i=2 \rightarrow (s=2(69\%) \vee s=3(31\%))$

与文献[16]中图4相比,同样是1层树,但其判定准确性(每项都有准确判定)高于文献[16]中的图4(以 $\lambda=0.72$,即全部判定可信度只有0.72的概率)。

对文献[17]中表2,首层划分属性*M*的选择:

$$U_{a_1} = \sum U_{a_{1v}} | (\forall v \in a_1) = \sum \frac{Y_{a_{1v}}}{X_{a_{1v}}} | (\forall v \in a_1) = \sum U_{a_{1v}} (v=1,2,3,4,5) =$$
$$\frac{Y_{a_{11}}}{X_{a_{11}}} + \frac{Y_{a_{12}}}{X_{a_{12}}} + \frac{Y_{a_{13}}}{X_{a_{13}}} + \frac{Y_{a_{14}}}{X_{a_{14}}} + \frac{Y_{a_{15}}}{X_{a_{15}}} = \frac{4}{2} + \frac{6}{3} + \frac{6}{2} + \frac{7}{3} + \frac{3}{2} \approx 10.8 \tag{7}$$

同理, $U_{a_2}=14, U_{a_3} \approx 12.3, U_{a_4}=14$ 。
由 $M=\max(U_{A_i}) \rightarrow U_{a_2}=14 (A_i=a_1, a_2, a_3, a_4)$ 。

经 $M=a_2$ 划分后最终得决策树:

$[a_2]: a_2=3 \rightarrow d=0(6)$
 $a_2=1$
 $[a_4]: a_4=3 \rightarrow d=2(1); a_4=4 \rightarrow d=1(2); a_4=5 \rightarrow d=2(1)$
 $a_4=1$
 $[a_3]: a_3=2 \rightarrow d=0(1); a_3=3 \rightarrow d=1(1)$
 $a_2=2$
 $[a_3]: a_3=2 \rightarrow d=0(1); a_3=3 \rightarrow d=0(1); a_3=5 \rightarrow d=1(1)$
 $a_3=1$
 $[a_4]: a_4=1 \rightarrow d=0(2); a_4=5 \rightarrow d=2(1)$
 $a_2=4$
 $[a_4]: a_4=2 \rightarrow d=2(1); a_4=4 \rightarrow d=2(2); a_4=5 \rightarrow d=1(2)$
 $a_2=5$
 $[a_1]: a_1=1 \rightarrow d=1(1); a_1=3 \rightarrow d=0(2)$

与文献[17]中图2相比,同样是3层树且最底层均为4叶结点,但本算法首层分辨出6行,而文献[17]的图2只分辨出4行。

3 结 论

- a. 从理论上证明非相容最小项集可经逻辑代换为等价的相容最小项集,使非相容表转换为相容表。
- b. 针对相容表的特点,提出了一种基于属性粒分辨关系的构建决策树算法。针对各类决策表,以条件属性集*A*中各属性粒对*D*的分辨关系为基础,不需属性约简直接确定划分属性*M*。
- c. 所提算法简洁易理解,时间和空间复杂度较小。
- d. 所提算法采用一级判定法确定*M*。由于决策树的多解性,若一级判定仍有多个等价的*M*时,可采用多级判定法确定*M*。

参考文献:

[1] 秦川,陈海军,施化吉,等. 不相容决策表的属性约简算法[J]. 计算机工程与应用,2008,44(24):162-164. (QIN Chuan, CHEN Haijun, SHI Huaji, et al. Attribute reduction algorithm for inconsistent decision tables[J]. Computer Engineering and Applications,2008,44(24):162-164. (in Chinese))

[2] 叶东毅. 不相容决策表分解的若干性质[J]. 小型微型计算机系统,2006,27(4):695-697. (YE Dongyi. Some properties of decomposition of inconsistent decision tables[J]. Journal of Chinese Computer Systems,2006,27(4):695-697. (in Chinese))

- [3] HUANG Longjun, ZHOU Caiying. A new method for constructing decision tree based on rough set theory[C]//Proceeding of International Conference on Granular Computing. Washington: IEEE,2007:241-244.
- [4] BAI J S, FAN B, XUE J Y. Knowledge representation and acquisition approach based on decision tree[C]//2003 IEEE International Conference on Natural Language Processing and Knowledge Engineering. Washington: IEEE,2003:533-538.
- [5] 王国胤,于洪,杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报,2002,25(7):759-766. (WANG Guoyin, YU Hong, YANG Dachun. Decision table reduction based on conditional information entropy[J]. Chinese Journal of Computers,2002,25(7):759-766. (in Chinese))
- [6] WANG Xiangyang, YANG Jie, JENSEN Richard, et al. Feature selection based on rough sets and particle swarm optimization[J]. Pattern Recognition Letters,2007,28(4):459-471.
- [7] JENSEN R, SHEN Q. Finding rough set reducts with ant colony optimization[C]//Proceedings of the 2003 UK Workshop on Computational Intelligence. Berlin: Springer,2003:15-22.
- [8] WANG GuoYin, ZHAO Jun, AN Jiujiang, et al. Theoretical study on attribute reduction of rough set theory: in algebra view and information view[C]//Third International Conference on Cognitive Informatics. Washington: IEEE,2004:148-155.
- [9] 李永华,蒋芸,王小菊. 一种基于 rough 集的属性约简的改进算法[J]. 计算机应用,2008,28(8):200-202. (LI Yonghua, JIANG Yun, WANG Xiaojun. An improved algorithm for attribute reduction based on rough sets[J]. Journal of Computer Applications,2008,28(8):200-202. (in Chinese))
- [10] 徐久成,史进玲,成万里. 粒计算中决策规则的提取[J]. 计算机工程与应用,2009,45(25):132-134. (XU Jiucheng, SHI Jinling, CHENG Wanli. Algorithm for decision rules extraction based on granular computing[J]. Computer Engineering and Applications,2009,45(25):132-134. (in Chinese))
- [11] 刘全,伏玉琛,孙吉贵,等. 一种基于集合符号的自动推理扩展方法[J]. 计算机研究与发展,2007,44(8):1317-1323. (LIU Quan, FU Yuchen, SUN Jigui, et al. An automated reasoning expanded method based on set signs[J]. Journal of Computer Research and Development,2007,44(8):1317-1323. (in Chinese))
- [12] 刘宏岚,高庆狮,杨炳儒. 多值逻辑中的命题相关性与逻辑运算研究[J]. 北京科技大学学报,2007,12(增刊2):177-182. (LIU Honglan, GAO Qingshi, YANG Bingru. Proposition relativity and logic calculation in many-valued logic [J]. Journal of University of Science and Technology Beijing,2007,12(Sup2):177-182. (in Chinese))
- [13] 王名扬,胡清华,于达仁. 基于粗糙集约简的决策林构建方法[J]. 计算机工程,2009,35(15):193-197. (WANG Mingyang, HU Qinghua, YU Daren. Construction method of decision forests based on rough set reduction[J]. Computer Engineering,2009,35(15):193-197. (in Chinese))
- [14] 朱承学,陈志刚. 不相容信息系统的属性约简研究[J]. 中南林业科技大学学报,2010,30(6):189-192. (ZHU Chengxue, CHEN Zhigang. Research of attributes reduction in inconsistent system[J]. Journal of Central South University of Forestry & Technology,2010,30(6):189-192. (in Chinese))
- [15] 王静,叶茂,刘启和,等. 布尔决策表的属性约简新方法:应用于欺诈识别[J]. 计算机工程与应用,2011,47(12):126-129. (WANG Jing, YE Mao, LIU Qihe, et al. New attribute reduction algorithm of Boolean decision table: application on fraud detection [J]. Computer Engineering and Applications,2011,47(12):126-129. (in Chinese))
- [16] 高静,徐章艳,宋威,等. 一种新的基于粗糙集模型的决策树算法[J]. 计算机工程,2008,34(3):9-11. (GAO Jing, XU Zhangyan, SONG Wei, et al. New decision tree algorithm based on rough set model[J]. Computer Engineering,2008,34(3):9-11. (in Chinese))
- [17] 周军,林庆. 基于粒度商的决策树构造算法[J]. 计算机工程与设计,2009,30(16):3826-3829. (ZHOU Jun, LIN Qing. Quotient GD based algorithm for design decision tree[J]. Computer Engineering and Design, 2009,30(16):3826-3829. (in Chinese))